TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Shape Optimization for a Variational Inequality: Comparison of a Regularized and a Unregularized Approach

ausgeführt am

Institut für
Analysis und Scientific Computing
TU Wien

unter der Anleitung von

Univ. Prof. Dipl.-Ing. Dr.techn.
Joachim Schöberl

und

Assistant Prof. Dipl.-Math. Dr.rer.nat.
Kevin Sturm

durch

Philipp Wörle, BSc.

Matrikelnummer: 01426582

Wien, am 16.12.2021

Joachim Schöberl
(Betreuer)

Philipp Wörle

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Formoptimierung hat zum Ziel eine Kostenfunktion, die in irgendeiner Weise von einer Menge Ω abhängig ist, zu optimieren. In dieser Arbeit ist die Kostenfunktion gegeben durch die quadrierte L^2-Norm von $y_\Omega - \overline{y}$ über einem Gebiet D für eine Zielfunktion $\overline{y}$. Hierbei ist y_Ω die Lösung einer Variationsungleichung, die von Ω abhängt und zusätzlich ein Hindernisproblem löst. Die Abhängigkeit der Variationsungleichung von Ω ist üblicherweise dadurch charakterisiert, dass die Variationsungleichung nur auf Ω gelöst wird. In dieser Arbeit ist allerdings die rechte Seite der Variationsungleichung von Ω abhängig. Dieses Problem wurde in der Literatur auf verschiedene Weise gelöst. Es werden zwei dieser Ansätze gegenübergestellt, wobei der Fokus auf die Unterschiede in der numerischen Lösung des Optimerungsproblems gelegt wird.

Im ersten Ansatz wird die Variationsungleichung regularisiert und dafür eine Formableitung berechnet. Die Formel wird anhand eines konkreten Beispiels numerisch getestet und mit Ergebnissen von Netgen/NGSolve, einer an der TU Wien entwickelten Software zum Lösen von Finite-Elemente-Problemen, verglichen. Im zweiten Ansatz wird das unregularisierte Problem direkt gelöst und ebenfalls eine Formableitung angegeben. Die numerische Berechnung dieser Formableitung stellt sich als schwieriger heraus. Es wird ein neuer Algorithmus entwickelt, der dies unter vereinfachten Annahmen an das Problem ermöglicht. Für den Algorithmus wird auch Konvergenz gezeigt. Es werden die numerischen Ergebnisse aus beiden Ansätzen verglichen und ein Dritter aufgezeigt, der Techniken aus den beiden Anderen kombiniert.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

The aim of shape optimization is to optimize a cost function which is in some way dependent on a set Ω. In this thesis the cost function is given by the squared L^2-Norm of $y_\Omega - \overline{y}$ over a domain D for some given target function $\overline{y}$. Here y_Ω is the solution of a variational inequality depending on Ω which in addition solves an obstacle problem. The dependence of the variational inequality on Ω usually is such that the variational inequality is solved only on Ω. In contrast in this thesis the right-hand side of the variational inequality is dependent of Ω. This problem has been solved in different ways in the literature. Two such approaches are compared with the focus lying on the differences in the numericals solutions of the optimization problem.

In the first approach the variational inequality is regularized and a shape derivative formula for the regularized problem is calculated. The formula is tested numerically on a concrete example and compared with the results from Netgen/NGSolve, a software that solves finite-element-problems developed at TU Wien. In the second approach the unregularized problem is solved directly and a shape derivative formula is stated. The numerical calculation of this shape derivative is more complex. A new algorithm is developed which allows the calculation under additional assumptions on the problem. For this algorithm convergence is shown. The numerical results of both approaches are compared and a third approach is shown, which combines techniques of the two former ones.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

I would like to express my sincere gratitude to Joachim Schöberl for supervising my thesis and especially to Kevin Sturm for his invaluable guidance and constant support throughout my work on this thesis. His lectures and seminars introduced me to this fascinating area of mathematics and I am sincerely grateful for that.

I would not have gotten so far in my studies without some very dear friends I met along the way. I want to thank Albert, Annika, Conrad, David, Gregor, Jakob, Markus and Max. You really made the ups of studying infinitely higher and the downs a mere cloudy moment.

As a student the offer of fresh, free and delicious food is one of the greatest things there are and Mona and Roel provided me with plenty throughout the years. I have the fondest memories of our lunches and dinners together. Thank you so much!

I also want to thank my siblings Paul, Maria, Judith, Christiane, Clemens and Anna for all the things great siblings do. You are the very best (usually)!

Last but not least I am deeply grateful to my parents. Without their selfless support thoughout my entire life I would not have been able to write these lines. Thank you for everything!

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 16.12.2021

Philipp Wörle

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

1. Introduction **1**

2. Introduction to Shape Optimization **5**
2.1. Notation 5
2.2. Shape Derivatives 6
2.3. Basic Results in Shape Optimization 9

3. A Fully Regularized Obstacle Problem **13**
3.1. Problem Definition 13
3.2. Regularizing the problem 14
3.3. Convergence of the Regularized Problems 17
3.4. A Shape Derivative Formula 24
3.4.1. On Lagrange Functions 24
3.4.2. The Averaged Adjoint Approach 26

4. Numerical Experiments for the Fully Regularized Problem **41**
4.1. Experiment setup 41
4.2. Experiments 43
4.2.1. The Target Function and Choice of $\overline{\lambda}$ 43
4.2.2. The Shape Derivative Formula 45
4.2.3. Shape Convergence 47

5. The Unregularized Obstacle Problem **53**
5.1. Abstract Operator Theory 53
5.2. Application of the Operator Theory 58

6. Numerical Experiments for the Unregularized Problem **71**
6.1. Experiment Setup 71
6.2. Solving the Obstacle Problem 72
6.3. Calculating the Material Derivative 74
6.4. Solving the Shape Optimization Problem 78
6.5. Comparing the Regularized with the Unregularized Problem 82

7. Final Remarks and Outlook **87**

Appendices **89**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

A. Code of Numerical Experiments **89**
A.1. Auxiliary Functions 89
A.2. Target function 93
A.3. Shape Derivative Formula 94
A.4. Shape Convergence 96
A.5. Solving the Obstacle Problem 99
A.6. Calculating the Material Derivative 100
A.7. Solving the Shape Optimization Problem 102
A.8. Comparing the Regularized with the Unregularized Problem 105

Bibliography **109**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1. Introduction

The aim of shape optimization is to optimize a function J with values in $\mathbb{R}$, called the cost function, depending on a set $\Omega \subset D \subset \mathbb{R}^n$ called the shape. D is a bounded set called the hold-all domain. The shape Ω usually is not completely arbitrary but has to be an element of $\mathcal{A} \subset 2^D$, a subset of the power set of D, called the admissible set. It greatly depends on the problem which form the cost function J, the hold-all domain D and the set of admissible sets take on.

One example is given by $J(\Omega) := \int_\Omega 1 \, \mathrm{d}x$ the volume of Ω subject to $\Omega \in \mathcal{A} := \{\Omega \subset D : \Omega \text{ is measurable, the surface } \partial\Omega \text{ is measurable and is equal to } 2\pi\}$, where $D = (0,2)^2$ and the goal is to maximize J. Therefore we look for the two-dimensional shape Ω with the largest area with a boundary of length 2π. The solution is of course the circle with radius 1, see [2, Beispiel 4.1].

In this thesis we look at the following problem. Given an open, bounded set D, a so called target function $\overline{y} \in H_0^1(D)$, a function $f_\Omega \in L^2(D)$ depending on Ω, an obstacle $\varphi \in L^1_{loc}(D)$ and a set $\mathcal{K}_\varphi := \{v \in H_0^1(D) : v(x) \leq \varphi(x) \text{ a.e. in } D\}$ we want to minimize

$$J(\Omega) := \int_D |y_\Omega - \overline{y}| \, \mathrm{d}x$$

subject to $y_\Omega \in H_0^1(D)$ solves the variational inequality

$$a(y_\Omega, v - y_\Omega) \geq (f_\Omega, v - y_\Omega)_{L^2(D)} \qquad \text{for all } v \in \mathcal{K}_\varphi$$

for some bilinear form $a(.,.)$ on $H^1(D)$.

The aim of this thesis is to compare two different approaches of solving this problem both in theory and numerically. The first approach transforms the variational inequality into a variational equality which is then regularized and solved. The second approach solves the unregularized variational inequality directly. We are particularly interested in the numerical aspects of these two approaches. How well does the regularized problem perform? Can we develop an algorithm to solve the unregularized shape optimization problem? Is convergence of such an algorithm guaranteed? How well does it perform compared to the regularized version?

Similar problems have also been studied by various other authors. The theoretical work of this thesis is mostly based on [11, 17] which is reviewed in greater detail

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

in this thesis. In [9] an algorithm for the same problem as discussed in this thesis was developed with the additional assumption of a two-valued, piece-wise constant function f_Ω. The same approach this thesis follows in Chapter 3, i.e. regularizing the problem and applying the averaged adjoint approach, can be found in [16] for a problem with applications in high-temperature superconductivity. In [12] the authors investigate the shape optimization problem subject to a variational inequality with a shape-independent right-hand side f. Many techniques used within this work can also be found in the context of optimal control problems, see e.g. [3, 13]. Introductions to shape optimization are found in [5, 19].

The outline of this thesis is as follows.

In Chapter 2 first the notation used throughout this thesis is introduced. Then a brief introduction on shape derivatives is given and some well known results in the theory of shape optimization are stated which will be used later on.

Chapter 3 begins with the definition of the shape optimization problem. Large parts of this chapter are based on [17]. Their regularization approach is reviewed and their convergence results for the solution of the regularized state equation is stated. A different convergence result for the adjoints is given, which is more in line with the literature, see [4]. The result in [17] on existence of the shape derivative via the average adjoint approach [21] is recited. Then a different way to calculate the shape derivative is given which requires less assumptions compared to the proof in [17].

In Chapter 4 similar experiments to the ones found in [17] are performed but the focus of the questions asked differs. Since at TU Wien a finite element software called Netgen/NGSolve is developed in which in recent versions a shape differentiation capability was added [10] all experiments compare the approach in Chapter 2 with the Netgen/NGSolve results. The shape optimization algorithm found in [10] is adapted to solve the regularized problem.

Chapter 5 lays the theoretical groundwork for the following chapter. The work in [11] is recited where the authors solved the unregularized problem in a more general form. Since the problem in this thesis is a special case the proofs are adapted and extended to allow for more general bilinear forms than the ones in [11].

In Chapter 6 a new algorithm to calculate the shape derivative for the unregularized problem is proposed based on the results of the previous chapter and using techniques from bound constraint problems found in [6]. A shape optimization algorithm is developed for which convergence is shown. Numerical experiments are conducted and the results are compared with the ones performed in Chapter 4. Con-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

vergence for an approach using the material derivative for the regularized problem with the techniques in this chapter is shown.

Chapter 7 concludes the thesis with some final remarks.

In the Appendix the code used in the experiments can be found.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2. Introduction to Shape Optimization

In this chapter we first introduce the notation used throughout this thesis. We then introduce the notion of shape derivatives and give a summary of basic definitions and results needed in this thesis.

2.1. Notation

D	bounded subset $D \subset \mathbb{R}^d$
$C^k(D)$	k-times continuously differentiable functions with values in $\mathbb{R}$
$C^{0,1}(\mathbb{R}^d, \mathbb{R}^d)$	Lipschitz continuous vector fields
$\mathring{C}^{0,1}(\overline{D}, \mathbb{R}^n)$	Lipschitz continuous vector fields vanishing on ∂D
$C_c^k(D, \mathbb{R}^d)$	k times continuously differentiable vector fields with compact support
$\|f\|_\infty$	$\sup_{x \in D} \|f(x)\|$
V'	dual space of the Banach space V
$\langle v, w \rangle_V$	natural pairing of $v \in V'$ and $w \in W$
$\overline{\Omega}$, $\partial\Omega$	closure and boundary of a set $\Omega \subset \mathbb{R}^d$
Ω	set $\Omega \subset D \subset \mathbb{R}^d$
Ω^c	complement of $\overline{\Omega}$ in D, i.e. $D \backslash \overline{\Omega}$
$\mathcal{P}(D)$	set of all subsets of D
$L^p(D)$	standard space of all measurable functions that are p-integrable for $1 \leq p < \infty$ on D
$\|f\|_{L^p(D)}$	$\left(\int_D \|f\|^p \, \mathrm{d}x\right)^{1/p}$
$(v, w)_{L^2(D)}$	standard $L^2(D)$ scalar product
$L^\infty(D)$	space of essentially bounded functions on D
$\|f\|_{L^\infty(D)}$	$\operatorname{ess\,sup}_{x \in D} \|f(x)\|$
$H^1(D)$	standard Sobolev space of one time weakly differentiable functions with weak derivative in $L^2(D)$
$H_0^1(D)$	functions in $H^1(D)$ where the Trace operator is 0
$\|f\|_{H_0^1(D)} = \|f\|_{H^1(D)}$	$\left(\sum_{\|\gamma\| \leq 2} \|\partial_\gamma f\|^2_{L^2(D)}\right)^{1/2}$
$(v, w)_{H_0^1(D)}$	standard $H_0^1(D)$ scalar product

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$\operatorname{div}(\theta)$	divergence of a vector field θ
∇f	gradient of a function or a vector field f
Δf	Laplacian of the function f
$\operatorname{inv}(M)$	inverse of M
$\det(M)$	determinant of M
$\operatorname{kern}(M)$	kernel of M
$\partial\Phi_t^\theta$	partial derivative of $\Phi_t^\theta = \Phi^\theta(t,x)$ w.r.t. x
$\mathbb{1}_\Omega$	indicator function of the set Ω
$\overline{\mathcal{F}}^V$	closure of a set of functions $\mathcal{F}$ in the Banach space V

2.2. Shape Derivatives

In this section we want to introduce the notion of shape derivatives, i.e. the derivatives of shape functions $\Omega \mapsto J(\Omega)$ where $\Omega \subset \mathbb{R}^d$ is a set and $J(\Omega) \in \mathbb{R}$. There are two major approaches on how to define the shape derivative in the literature. One is the *method of perturbation of the identity* and the other one is the *velocity method*. We will introduce and use the latter one throughout this thesis. For more information on the former one and a comparison between those two methods we refer to [5].

To define a shape derivative we need to somehow perturbe our shape Ω. We start by introducing the theoretical framework to be able to perturbate a set $\Omega \subset \mathbb{R}^d$ using a vector field.

Definition 2.1. A subset $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is called *path connected*, if for any two points $a, b \in D$ there is a continuos map $p : [0,1] \to D$ such that $p(0) = a$ and $p(1) = b$.

Definition 2.2. A subset $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is called *simply connected* if it is path connected and for any two continuous paths $p : [0,1] \to D$, $q : [0,1] \to D$ with $p(0) = q(0)$ and $p(1) = q(1)$ there exists a continuous map $F : [0,1] \times [0,1] \to D$ such that $F(x,0) = p(x)$ and $F(x,1) = q(x)$. The map F is called a *homotopy*.

To put Definition 2.2 in other words, our subset D is not allowed to contain any holes.

Definition 2.3. A subset $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is called a *regular domain* if it is a simply connected and bounded domain with Lipschitz boundary $\Sigma := \partial D$.

Due to the Theorem of Picard-Lindelöf we have well-definedness of the following definition.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Definition 2.4. Let $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be a regular domain and $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d) := \left\{\theta|_{\overline{D}} : \theta \in C^{0,1}(\mathbb{R}^d, \mathbb{R}^d),\ \theta = 0 \text{ on } \partial D\right\}$ be a Lipschitz continuous vector field. Let $\tau > 0$ be arbitrary and α_{x_0} be the unique solution of

$$\alpha'_{x_0}(t) = \theta(\alpha_{x_0}(t)) \quad \text{for all } t \in [-\tau, \tau], \qquad \alpha_{x_0}(0) = x_0.$$

Then the *flow of* θ is defined by $\Phi_t^\theta(x_0) := \alpha_{x_0}(t)$.

Theorem 2.5 (Kirszbraun theorem)**.** *Let $\Omega \subset \mathbb{R}^n$ and $\theta : \Omega \to \mathbb{R}^d$ be a Lipschitz-continuous function with Lipschitz constant L. Then there exists a Lipschitz-continuous function $\widetilde{\theta} : \mathbb{R}^d \to \mathbb{R}^d$ with the same Lipschitz constant L such that $\widetilde{\theta}\big|_\Omega = \theta$.*

Proof. See [8]. □

At this point we want to give a motivation for the choice of $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$ instead of $\theta \in C^{0,1}(\overline{D}, \mathbb{R}^d)$. On the one hand using Kirszbaum theorem we can extend both $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$ and $\theta \in C^{0,1}(\overline{D}, \mathbb{R}^d)$ to global Lipschitz continuous vector fields $\widetilde{\theta} \in C^{0,1}(\mathbb{R}^d, \mathbb{R}^d)$ such that $\widetilde{\theta}|_D = \theta$ and hence again using Picard-Lindelöf obtain a global solution $\alpha_{x_0} : \mathbb{R} \to \mathbb{R}^d$ and therefore a global flow $\Phi : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. On the other hand $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$ guarantees us strict θ-flow invariance of $\overline{D}$ where we call a set $D \subset \mathbb{R}^d$ *strictly θ-flow invariant* if $\Phi_t^\theta(\overline{D}) = \overline{D}$. More precisely we have the following result.

Theorem 2.6. *Let $D \subset \mathbb{R}^d$ be a regular domain, $\tau > 0$, and let $\theta \in C^{0,1}(\overline{D}, \mathbb{R}^d)$ satisfy for all $x \in \overline{D}$ that $\pm\theta(x) \in T_{\overline{D}}(x)$ where*

$$T_{\overline{D}}(x) := \left\{h \in \mathbb{R}^d : (x_\nu - x)/\tau_\nu \to h \ \text{ for some } \ x_\nu \to_{\overline{D}} x, \tau_\nu \to 0\right\}.$$

Then D is strictly θ-flow invariant and it holds for some constants $C, c > 0$:

$$\begin{aligned} \forall x \in \overline{D} \ : \ & t \mapsto \Phi_t^\theta(x) \in C^1([0,\tau], \mathbb{R}^d), \\ & t \mapsto \left(\Phi_t^\theta\right)^{-1}(x) \in C([0,\tau], \mathbb{R}^d), \end{aligned}$$

$$\begin{aligned} \forall x, y \in \overline{D} \ : \ & \|\Phi^\theta(x) - \Phi^\theta(y)\|_{C^1([0,\tau],\mathbb{R}^d)} \le C|x-y|, \\ & \|\left(\Phi^\theta\right)^{-1}(x) - \left(\Phi^\theta\right)^{-1}(y)\|_{C([0,\tau],\mathbb{R}^d)} \le c|x-y| \end{aligned}$$

and for all $t \in \mathbb{R}$

$$\Phi_t^\theta(\mathrm{int}(D)) = \mathrm{int}(D) \quad \textit{and} \quad \Phi_t^\theta(\partial D) = \partial D,$$

where $\mathrm{int}(D)$ *denotes the interior of D.*

Proof. See [20, Theorem 2.10]. □

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

To see that for $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$ all conditions in Theorem 2.6 are fullfilled take for $x \in \operatorname{int}(D)$ some $h \in \mathbb{R}^d$. Since $x \in \operatorname{int}(D)$ is open there exists a $\tau_0 > 0$ small enough, such that $x + \tau h \in \operatorname{int}(D)$ for all $0 < \tau < \tau_0$. Let $(\tau_\nu)_\nu$ be a sequence such that $\tau_\nu \to 0$ and let $x_\nu := x + \tau_\nu h$. Then $x_\nu \to_{\overline{D}} x$ and $(x_\nu - x)/\tau_\nu = h \to h$. Since $h \in \mathbb{R}^d$ was arbitrary it follows that $T_{\overline{D}}(x) = \mathbb{R}^d$ for $x \in \operatorname{int}(D)$. For $x \in \partial D$ we note that $\theta(x) = 0$ for $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$. To see $0 \in T_{\overline{D}}(x)$ we can take any positive sequence $(\tau_\nu)_\nu$ and $(x_\nu)_\nu := (x)_\nu$.

We now want to introduce shape functions and their derivatives.

Definition 2.7. Let $D \subset \mathbb{R}^d$ be a set, $\mathcal{P}(D) := \{\Omega : \Omega \subset D\}$ and $\mathcal{A} \subset \mathcal{P}(D)$ be a set of subsets. Then a function

$$J : \mathcal{A} \to \mathbb{R} : \quad \Omega \mapsto J(\Omega)$$

is called a *shape function*. $\mathcal{A}$ is called the *admissible set*.

In shape optimization we usually want to find a minimizer of the problem

$$\min_{\Omega \in \mathcal{A}} J(\Omega).$$

A typical example for J is $J(\Omega) := \int_\Omega f \, \mathrm{d}x$ for some $f \in L^1(D)$. Throughout this thesis our shape function J will also depend implicitly on a solution of a PDE. We want to give a brief example for such a problem.

Example 2.8. Let $D \subset \mathbb{R}^d$ be an open and bounded domain, $\Omega \subset D$ be open and $\overline{y} \in L^1(D)$, $f \in L^2(D)$ be given functions. Let $y_\Omega \in H_0^1(\Omega)$ be the solution of the Poisson equation

$$\begin{cases} -\Delta y_\Omega = f, & \text{in } \Omega, \\ y_\Omega = 0, & \text{on } \partial\Omega. \end{cases}$$

Our PDE-dependent shape function J is given by

$$J(\Omega) := \int_\Omega |y_\Omega - \overline{y}| \, \mathrm{d}x.$$

In a real world application $\overline{y}$ could be some measured data. By minimizing $J(\Omega)$ we would then find an $\Omega \subset D$ and a solution y_Ω, such that our desired $\overline{y}$ is approximated best by y_Ω in $L^1(D)$.

We now want to define the derivative of a shape function J in direction of some vector field θ.

Definition 2.9. Let $D \subset \mathbb{R}^d$ be a regular domain, $\Omega \subset D$ and J be a shape function. Let $\theta \in \mathring{C}^{0,1}(\overline{D}, \mathbb{R}^d)$ be a vector field and $\Omega_t := \Phi_t^\theta(\Omega)$. Then the *Eulerian semi-derivative* of J at Ω in direction of θ is defined by

$$\mathrm{d}J(\Omega)(\theta) := \lim_{t \to 0+} \frac{J(\Omega_t) - J(\Omega)}{t},$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

if this limit exists. J is said to be *shape differentiable* at Ω if it has an Eulerian semi-derivative at Ω for all $\theta \in C_c^\infty(\overline{D}, \mathbb{R}^d)$ and the mapping

$$\mathrm{d}J(\Omega) : C_c^\infty(\overline{D}, \mathbb{R}^d) \to \mathbb{R} : \quad \theta \mapsto \mathrm{d}J(\Omega)(\theta)$$

is linear and continuous, in which case $\mathrm{d}J(\Omega)(\theta)$ is called the *shape derivative* at Ω.

2.3. Basic Results in Shape Optimization

We now want to state some well-known results in shape optimization which we use later on.

Theorem 2.10. *Let $D \subset \mathbb{R}^d$ be a regular domain. Let $\theta \in C_c^1(D, \mathbb{R}^d)$ and Φ_t^θ be its flow. Let $f \in L^2(D)$. Then we have*

$$\lim_{t \to 0^+} \|f \circ \Phi_t^\theta - f\|_{L^2(D)} = 0 \qquad \textit{and} \qquad \lim_{t \to 0^+} \|f \circ \left(\Phi_t^\theta\right)^{-1} - f\|_{L^2(D)} = 0.$$

Proof. See [5, Chapter 11, Lemma 2.1]. □

Theorem 2.11. *Let $D \subset \mathbb{R}^d$ be a regular domain. Let $\theta \in C_c^1(D, \mathbb{R}^d)$ and Φ_t^θ be its flow. Let $f \in H_0^1(D)$. Then we have*

$$\lim_{t \to 0^+} \|f \circ \Phi_t^\theta - f\|_{H_0^1(D)} = 0.$$

Proof. See [20, Lemma 2.16 (ii)]. □

The proof of this theorem makes use of the following identity which is an immediate result of applying the chain rule on the left-hand side,

$$\nabla \left(f \circ \Phi_t^\theta\right) = \left(\partial \Phi_t^\theta\right)^T \nabla f \circ \Phi_t^\theta.$$

This identity is used a lot throughout this thesis in the rearranged form

$$\nabla f \circ \Phi_t^\theta = \left(\partial \Phi_t^\theta\right)^{-T} \nabla \left(f \circ \Phi_t^\theta\right).$$

Theorem 2.12. *Let $D \subset \mathbb{R}^d$ be a regular domain. Let $\theta \in C_c^1(D, \mathbb{R}^d)$ and Φ_t^θ be its flow. Let $f \in H_0^1(D)$. Then we have*

$$\lim_{t \to 0^+} \left\| \frac{f \circ \Phi_t^\theta - f}{t} - \nabla f \cdot \theta \right\|_{L^2(D)} = 0.$$

Proof. See [19, Proposition 2.37]. □

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Theorem 2.13. *Let $D \subset \mathbb{R}^d$ be a regular domain. Let $\theta \in C_c^1(D, \mathbb{R}^d)$ and Φ_t^θ be its flow. Let $\tau > 0$ and $(f_t)_t \in H_0^1(D)$ be a family of funcions for $t \in [0, \tau]$. Let $t \mapsto f_t : [0, \tau] \to H_0^1(D)$ be a continuous function. Then $t \mapsto f_t \circ \Phi_t^\theta : [0, \tau] \to H_0^1(D)$ is a continuous function and we have*

$$\lim_{t \to 0^+} \left\| f_t \circ \Phi_t^\theta - f \right\|_{H_0^1(D)} = 0.$$

Proof. See [20, Lemma 2.16 (iv)]. □

We perform several integrations by substitution later on for which the following results will be of great use.

Theorem 2.14. *Let $D \subset \mathbb{R}^d$ be an open and bounded domain and $\theta \in C_c^1(D, \mathbb{R}^d)$. Then we have*

$$\lim_{t \to 0^+} \frac{\partial \Phi_t^\theta - \mathrm{Id}}{t} \to \nabla \theta \qquad \textit{strongly in } C(\overline{D}, \mathbb{R}^{d,d}),$$

$$\lim_{t \to 0^+} \frac{\det(\partial \Phi_t^\theta) - 1}{t} \to \mathrm{div}(\theta) \qquad \textit{strongly in } C(\overline{D}).$$

Proof. See [19, Proposition 2.44]. □

We also need the derivative of $\left(\partial \Phi_t^\theta\right)^{-1}$ with respect to t at $t = 0$. To calculate it the following lemma will be of great use.

Theorem 2.15. *Let GL_d be the set of invertible matrices $A \in \mathbb{R}^{d,d}$. Then the mapping $\mathrm{inv}(A) := A^{-1} : \mathrm{GL}_d \to \mathbb{R}^{d,d}$ is differentiable and its differential is given by*

$$\partial \mathrm{inv}(A)(B) = -A^{-1} B A^{-1}, \qquad \textit{for all } A \in \mathrm{GL}_d,\ B \in \mathbb{R}^{d,d}.$$

Proof. See [1, Satz 7.2 (ii)]. □

Theorem 2.16. *Let $D \subset \mathbb{R}^d$ be an open and bounded domain and $\theta \in C_c^1(D, \mathbb{R}^d)$. Then we have*

$$\lim_{t \to 0^+} \frac{\left(\partial \Phi_t^\theta\right)^{-1} - \mathrm{Id}}{t} \to -\nabla \theta \qquad \textit{strongly in } C(\overline{D}, \mathbb{R}^{d,d}).$$

Proof. Using the chainrule, $\partial \Phi_0^\theta = \mathrm{Id}$ and Theorem 2.14 we have

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \left(\partial \Phi_t^\theta\right)^{-1} \Big|_{t=0} &= \frac{\mathrm{d}}{\mathrm{d}t} \mathrm{inv}\left(\partial \Phi_t^\theta\right) \Big|_{t=0} \\ &= \partial \mathrm{inv}\left(\partial \Phi_0^\theta\right) \left(\frac{\mathrm{d}}{\mathrm{d}t} \partial \Phi_t^\theta \Big|_{t=0} \right) \\ &= \partial \mathrm{inv}\left(\mathrm{Id}\right) \left(\nabla \theta\right). \end{aligned}$$

We can now apply Theorem 2.15 and get

$$\partial \mathrm{inv}\left(\mathrm{Id}\right) \left(\nabla \theta\right) = -\mathrm{Id}^{-1} \left(\nabla \theta\right) \mathrm{Id}^{-1} = -\nabla \theta.$$

□

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The following theorem is a generalization of [11, Lemma 9 (i)].

Theorem 2.17. *Let $D \subset \mathbb{R}^d$ be an open, bounded domain and let $\theta \in C_c^1(D, \mathbb{R}^d)$. Let $M = (m_{i,j})_{i,j}$ with $m_{i,j} \in C^1(\overline{D})$ for $i, j = 1, ..., d$ and $\xi(t) = \det\left(\partial\Phi_t^\theta\right)$. Define*

$$M(t) := \xi(t)(\partial\Phi_t)^{-1}(M \circ \Phi_t)(\partial\Phi_t)^{-T}.$$

Then the following convergences hold true

$$\frac{M(t) - M}{t} \to M'(0) = \operatorname{div}(\theta)M - \nabla\theta M + \nabla M \cdot \theta - M(\nabla\theta)^T$$

strongly in $C(\overline{D}, \mathbb{R}^{d,d})$, where $\nabla M \cdot \theta$ denotes the matrix $(\nabla m_{i,j}^T \theta)_{i,j}$.

Let furthermore M be positive definite. Then there is also a constant $t^* > 0$ such that for all $t \in [0, t^*]$, for all $x \in \overline{D}$ and for all $p \in \mathbb{R}^d$

$$M(t,x)p \cdot p \geq \frac{1}{2}p^T Mp,$$
$$\xi(t,x) \geq \frac{1}{2}.$$

Proof. Applying the chainrule three times and using the theorems in this section leads to the four terms. The inequalities directly follow from the continuity of $M(t)$ and $\xi(t)$ at $t = 0$. □

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3. A Fully Regularized Obstacle Problem

In this chapter we review the work done in [17]. We start by formulating the problem discussed by [17] and restate the main results. We prove a different result for the convergence of the adjoints. We then give a short introduction on Lagrangians on Banach spaces. We conclude this chapter by using an easier and more direct approach to prove an explicit shape derivative formula for our problem compared to the one found in [17].

3.1. Problem Definition

Let $D \subset \mathbb{R}^d$ be a bounded domain equipped with a sufficiently smooth boundary ∂D.

We assume that our hold-all domain D is partitioned into three disjoint subdomains Ω, Γ and Ω^c, where Ω, Ω^c are open, $\overline{\Omega}$ is a proper subset of D, $\partial\Omega = \Gamma$ and $\Omega^c = D \backslash \overline{\Omega}$. In other words we can write D as the disjoint union $D = \Omega \dot{\cup} \Gamma \dot{\cup} \Omega^c$.

Let $f_\Omega \in L^2(D)$ be dependent on the shape Ω and let an obstacle $\varphi \in L^1_{loc}(D)$ be such that the set of admissible functions

$$\mathcal{K}_\varphi := \left\{\theta \in H^1_0(D) : \theta(x) \leq \varphi(x) \ \text{ a.e. in } D\right\}$$

is not empty. Furthermore given $M := (m_{i,j})_{i,j}$, $e := (e_i)_i$ and b with $m_{i,j}$, e_i, $b \in L^\infty(D)$, $i, j = 1, ..., d$, let

$$\begin{aligned} a : H^1_0(D) \times H^1_0(D) &\to \mathbb{R} \\ (y, v) &\mapsto \int_D \sum_{i,j=1}^d m_{i,j}\partial_i y \partial_j v + \sum_{i=1}^d e_i(\partial_i y v + y\partial_i v) + byv \ \mathrm{d}x \\ &= \int_D \nabla y^T M \nabla v + e^T(v\nabla y + y\nabla v) + byv \ \mathrm{d}x \end{aligned}$$

be a bounded, coercive and symmetric bilinear form. Let $y_\Omega \in \mathcal{K}_\varphi$ be the unique solution of the variational inequality

$$a(y_\Omega, v - y_\Omega) \geq (f_\Omega, v - y_\Omega)_{L^2(D)} \ \text{ for all } v \in \mathcal{K}_\varphi. \tag{3.1}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Existence and uniqueness of y_Ω in our setting follow from [15, Theorem 2.1].

Finally let $\overline{y} \in H^1(D)$. Then our shape function $J(\Omega)$ is defined by

$$J(\Omega) := \frac{1}{2} \int_D |y_\Omega - \overline{y}|^2 \, \mathrm{d}x \tag{3.2}$$

and the shape optimization problem is given by

$$\min_{\Omega \in \mathcal{A}} J(\Omega), \tag{3.3}$$

where $\mathcal{A}$ is some set of admissible shapes.

Let us take a closer look at our setting. We want to emphasize how our shape function depends on the shape Ω. Unlike in Example 2.8 the solution y_Ω of (3.1) is defined on the hold-all domain D. The shape dependency of y_Ω stems from f_Ω in that equation. In our numerical experiments later on we will consider the case of a piecewise constant function f_Ω where f_Ω takes one value on Ω and another one on Ω^c.

3.2. Regularizing the problem

In this section we are going to regularize (3.1). We get two kinds of regularization which in [17] are called the *regularized state equation* and the *fully regularized state equation*. With every regularization there comes a trade-off. On the one hand we are able to apply more sophisticated theory to our problem, on the other hand we have to show convergence of the regularized problem to the unregularized one.

As a first step, we bring inequality (3.1) into an equality form as we have the following result.

Theorem 3.1. *Given $\varphi \in L^2(D)$. Let $(y_\Omega, \lambda) \in H_0^1(D) \times L^2(D)$ solve*

$$a(y_\Omega, v) + (\lambda, v)_{L^2(D)} = (f_\Omega, v)_{L^2(D)} \quad \forall v \in H_0^1(D), \tag{3.4}$$

$$\begin{aligned} \lambda \geq 0 \quad & a.e. \text{ in } D, \\ y_\Omega \leq \varphi \quad & a.e. \text{ in } D, \\ \lambda(y_\Omega - \varphi) = 0 \quad & a.e. \text{ in } D. \end{aligned} \tag{3.5}$$

Then $y_\Omega \in \mathcal{K}_\varphi \subset H_0^1(D)$ solves (3.1).

Indeed also the reverse is true, i.e. if $(y_\Omega, \lambda) \in H_0^1(D) \times L^2(D)$ solves (3.4) and (3.5) y_Ω also solves (3.1). Since we only need one direction in this thesis we may refer the interested reader to [15] for the proof of the equivalence.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. Since $v - y_\Omega \in H_0^1(D)$ for every $v \in H_0^1(D)$ it also holds that

$$a(y_\Omega, v - y_\Omega) + (\lambda, v - y_\Omega)_{L^2(D)} = (f_\Omega, v - y_\Omega)_{L^2(D)} \tag{3.6}$$

for all $v \in \mathcal{K}_\varphi \subset H_0^1(D)$. Due to (3.5) we have $(\lambda, v - y_\Omega)_{L^2(D)} \leq 0$ since for $v \in \mathcal{K}_\varphi$ either $\lambda = 0$ where $y_\Omega \neq \varphi$ or $v - y_\Omega < 0$ where $y_\Omega = \varphi$. So by dropping the term $(\lambda, v - y_\Omega)_{L^2(D)}$ in (3.6) we get (3.1). □

We now observe that the three equations in (3.5) are equivalent to

$$\lambda = \max(0, \lambda + c(y_\Omega - \varphi)) \quad \text{for every } c > 0. \tag{3.7}$$

We give a brief motivation for why this is true. Note that the following arguments hold true almost everywhere.

First let (3.5) hold and let x be such that $\lambda(x) = 0$. Since $y_\Omega(x) \leq \varphi(x)$ we have $c(y_\Omega(x) - \varphi(x)) \leq 0$ for all $c > 0$ an therefore

$$0 = \lambda(x) = \max(0, 0 + c(y_\Omega(x) - \varphi(x))).$$

Now let x be such that $\lambda(x) > 0$. Then $y_\Omega(x) - \varphi(x) = 0$ due to $\lambda(x)(y_\Omega(x) - \varphi(x)) = 0$ and therefore

$$\lambda(x) = \max(0, \lambda(x) + 0) = \max(0, \lambda(x) + c(y_\Omega(x) - \varphi(x))).$$

To show the reverse let (3.7) hold. It immediately follows that $\lambda \geq 0$ in D. For every $x \in D$ we set $c := \lambda(x)$ in (3.7) and get

$$\lambda(x) = \max(0, \lambda(x) + \lambda(x)(y_\Omega(x) - \varphi(x))).$$

Therefore $\lambda(y_\Omega - \varphi) = 0$ in D. Finally suppose there was an $x \in D$ such that $y_\Omega(x) > \varphi(x)$. Then $\lambda(x) = \max(0, \lambda(x) + c(y_\Omega(x) - \varphi(x)))$ would be a contradiction.

According to [13, p.42] simply substituting (3.7) in (3.4) leads to serious difficulties numerically and in terms of convergence analysis of numerical implementations. In a numerical algorithm we would approximate $\lambda \in L^2(D)$ in (3.7) step-wise by some $\lambda_{k+1} \in H^{-1}(D)$. Unfortunately though, λ_{k+1} unlike λ is not an element of $L^2(D)$ but instead of $H^{-1}(D)$, the dual space of $H_0^1(D)$. Therefore the authors of [13] proposed a relaxation of (3.7) of the form

$$\lambda = \max(0, \overline{\lambda} + c(y_\Omega - \varphi)), \quad 0 < c < +\infty, \tag{3.8}$$

for some $\overline{\lambda}$, which was picked up in [17]. By putting (3.8) into (3.4) we get our *regularized state equation*

$$a(y_{c,\Omega}, v) + (\max(0, \overline{\lambda} + c(y_{c,\Omega} - \varphi)), v)_{L^2(D)} = (f_\Omega, v)_{L^2(D)} \quad \forall v \in H_0^1(D). \tag{3.9}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

At this point a few remarks about $\overline{\lambda}$ are necessary. A key notion is that we can choose $\overline{\lambda}$ arbitrarily and therefore assume more regularity, for example $\overline{\lambda} \in H^1(D)$. This is of particular importance when we derive a shape derivative formula for (3.2). The solution $(y_{c,\Omega}, \lambda_c)$ of

$$\begin{cases} a(y_{c,\Omega}, v) + (\lambda_c, v)_{L^2(D)} = (f_\Omega, v)_{L^2(D)} \quad \forall v \in H_0^1(D) \\ \lambda_c = \max(0, \overline{\lambda} + c(y_{c,\Omega} - \varphi)) \end{cases} \tag{3.10}$$

converges to the solution (y_Ω, λ) of (3.4), (3.5) as $c \to \infty$. A proof for this statement can be found in [13, Theorem 3.1] which we also restate for sake of completeness in Theorem 3.5. We note that the function $y_{c,\Omega}$ might not fulfill $y_{c,\Omega} \leq \varphi$. To achieve convergence from the correct direction, i.e. $y_{c_1,\Omega} \leq y_{c_2,\Omega} \leq y_\Omega \leq \varphi$ for $0 < c_1 < c_2$, a sufficient criterium is given in [13, (3.10)]. This criterium is satisfied by choosing $\overline{\lambda} = \max(0, f_\Omega - \mathbf{S}\varphi)$ where $\mathbf{S}$ is the differential operator corresponding to $a(.,.)$.

To get the fully regularized state equation we replace the max-function by a regularized $\max_\gamma$ which fulfills the following assumptions.

Assumption 3.2.

(i) $\max_\gamma \in C^1(\mathbb{R})$ *for all* $\gamma > 0$.

(ii) There exists a function $g : (0, +\infty) \to [0, +\infty)$ *with* $g(\gamma) \to 0$ *as* $\gamma \to +\infty$, *such that* $|\max_\gamma(x) - \max(0, x)| \leq g(\gamma)$ *for all* $x \in \mathbb{R}$ *and for all* $\gamma > 0$.

(iii) For all $x \in \mathbb{R}$ *and all* $\gamma > 0$ *we have* $\max'_\gamma(x) \in [0, 1]$ *and* $\max'_\gamma$ *is monotonically nondecreasing.*

(iv) Let $\delta > 0$. *Then* $\max'_\gamma$ *converges uniformly to 0 on* $(-\infty, -\delta)$ *and to 1 on* $(\delta, +\infty)$ *as* $\gamma \to +\infty$.

Thus we get the *fully regularized state equation*

$$a(y_{\gamma,c,\Omega}, v) + (\max_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)), v)_{L^2(D)} = (f_\Omega, v)_{L^2(D)} \quad \forall v \in H_0^1(D). \tag{3.11}$$

Closely related to our problem is the so called adjoint equation. In our setting it is given by: Find $p_{\gamma,c,\Omega} \in H_0^1(D)$ such that for all $v \in H_0^1(D)$ it holds that

$$a(p_{\gamma,c,\Omega}, v) + c(\max'_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))p_{\gamma,c,\Omega}, v)_{L^2(D)} = -(y_{\gamma,c,\Omega} - \overline{y}, v)_{L^2(D)}. \tag{3.12}$$

A motivation for where this equation comes from and why we are so interested in its solution and behavior will be given in Sections 3.4.1 and 3.4.2.

So far we have only considered the case in which our solution y_Ω satisifes $y_\Omega \leq \varphi$. If we had the reverse obstacle problem, i.e. $\varphi \leq y_\Omega$ all results throughout this thesis remain valid, only the places of those two functions in (3.7) would swap and we would have to change some signs accordingly.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.3. Convergence of the Regularized Problems

In this section we show the convergences of $y_{\gamma,c,\Omega} \to y_{c,\Omega}$ and $y_{c,\Omega} \to y_\Omega$ as $\gamma \to +\infty$ and $c \to +\infty$ respectively. We also state our own result for the convergence of $p_{\gamma,c,\Omega}$. A proof for the first convergence result for our scenario was first given in [17, Theorem 3.3] and for the second convergence result in [13, Theorem 3.1].

For the following proof we will make use of Nemytskii operators and the Browder-Minty theorem.

Definition 3.3. Let $D \subset \mathbb{R}^d$ be a domain. We say that a function

$$m \ : \ D \times \mathbb{R}^m \to \mathbb{R} \ : \ (x, y) \mapsto m(x, y)$$

satisfies the *Carathéodory conditions* if

- $y \mapsto m(x, y)$ is continuous for almost every $x \in D$ and
- $x \mapsto m(x, y)$ is measurable for every $y \in \mathbb{R}^d$.

Given a function m satisfying the Carathéodory conditions and a function $y : D \to \mathbb{R}^m$ we define the *Nemytskii operator* $\widetilde{\boldsymbol{M}}$ by composition

$$\widetilde{\boldsymbol{M}}(y)(x) := m(x, y(x)), \quad x \in D.$$

Theorem 3.4 (Browder-Minty)**.** *Let X be a real, reflexive Banach space and let $\mathbf{T} : X \to X'$ be bounded, continuous, coercive and strictly monotone. Denote by X' the dual space of X. Then for any $f \in X'$ there exists a unique solution y of the equation*

$$\mathbf{T}(y) = f.$$

A proof of this theorem can be found in [18, Section 3.1.1]. We can now state the first main theorem which is largely based on [13, Theorem 3.1] and [17, Proposition 3.1].

Theorem 3.5. *Let $D \subset \mathbb{R}^d$ be an open, bounded domain with Lipschitz boundary ∂D, the bilinear form $a(.,.)$ be as in Section 3.1, $f_\Omega \in L^2(D)$, $c > 0$, $\varphi \in H^1(D)$ and $\overline{\lambda} \in L^2(D)$. Let y_Ω be the solution to (3.4). Then (3.9) admits a unique solution $y_{c,\Omega}$ and*

$$\begin{aligned} y_{c,\Omega} &\to y_\Omega && \text{in } H^1(D), \\ \lambda_c &\rightharpoonup \lambda && \text{weakly in } H^{-1}(D) \end{aligned}$$

as $c \to +\infty$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. We start by showing the existence of a unique solution of (3.10). We define the function

$$m : D \times \mathbb{R} : (x, y) \mapsto \max(0, \overline{\lambda}(x) + c(y - \varphi(x))),$$

which satisifies the Carathéodory conditions. The associated Nemytskii operator $\widetilde{\boldsymbol{M}} : H^1(D) \to L^2(D)$ is monotone, since it is the composition of the monotone operator

$$\mathbf{max} : H_0^1(D) \to H_0^1(D) : y \mapsto \max(0, y)$$

and the affine linear operator

$$\boldsymbol{\Psi_c} : H_0^1(D) \to L^2(D) : y \mapsto \overline{\lambda} + c(y - \varphi).$$

We note, that $\widetilde{\boldsymbol{M}}$ is also continuous and defines an operator $\mathbf{M} : H_0^1(D) \to L^2(D) \subset H^{-1}(D)$ by

$$\langle \mathbf{M}y, v\rangle_{H^{-1}(D)\times H_0^1(D)} := (\widetilde{\boldsymbol{M}}(y), v)_{L^2(D)} \qquad \text{for all } v \in H_0^1(D).$$

Let the operator $\mathbf{A} : H_0^1(D) \to H^{-1}(D)$ be defined by

$$\langle \mathbf{A}y, v\rangle_{H^{-1}(D)\times H_0^1(D)} = a(y, v) \qquad \text{for all } v \in H_0^1(D).$$

The operator $\mathbf{A}$ is strictly monotone, coercive and continuous. Therefore $\mathbf{T} := \mathbf{A} + \mathbf{M} : H_0^1(D) \to H^{-1}(D)$ is also strictly monotone, coercive and continuous with the same coercivity constant $\mu > 0$ as $\mathbf{A}$. Monotonicity and continuity of $\mathbf{T}$ follow directly from monotonicity and continuity of $\mathbf{A}$ and $\mathbf{M}$. Thus we can apply the theorem of Browder-Minty and obtain a unique solution $y_{c,\Omega}$ for every $c > 0$ and $f_\Omega \in L^2(D)$.

We now show the convergence of $y_{c,\Omega}$ to the solution of the unregularized problem. Due to $\lambda_c \geq 0$, $c > 0$ and $\varphi - y_\Omega \geq 0$ we note that

$$\begin{aligned}
(\lambda_c, y_{c,\Omega} - y_\Omega)_{L^2(D)} &= \left(\lambda_c, \left(\frac{\overline{\lambda}}{c} + y_{c,\Omega} - \varphi\right) + \underbrace{\varphi - y_\Omega}_{\geq 0} - \frac{\overline{\lambda}}{c}\right)_{L^2(D)} \\
&\geq \frac{1}{c}\left(\lambda_c, \overline{\lambda} + c(y_{c,\Omega} - \varphi)\right)_{L^2(D)} - \frac{1}{c}\left(\lambda_c, \overline{\lambda}\right)_{L^2(D)} \\
&= \frac{1}{c}\|\lambda_c\|^2_{L^2(D)} - \frac{1}{c}\left(\lambda_c, \overline{\lambda}\right)_{L^2(D)}.
\end{aligned} \tag{3.13}$$

Since y_Ω, $y_{c,\Omega} \in H_0^1(D)$ we can use $y_{c,\Omega} - y_\Omega$ as a test function in (3.10) to obtain

$$a(y_{c,\Omega}, y_{c,\Omega} - y_\Omega) + (\lambda_c, y_{c,\Omega} - y_\Omega)_{L^2(D)} = (f_\Omega, y_{c,\Omega} - y_\Omega)_{L^2(D)}$$

or equivalently

$$a(y_{c,\Omega}, y_{c,\Omega}) + (\lambda_c, y_{c,\Omega} - y_\Omega)_{L^2(D)} = a(y_{c,\Omega}, y_\Omega) + (f_\Omega, y_{c,\Omega} - y_\Omega)_{L^2(D)}.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Putting inequality (3.13) into this equation we get

$$a(y_{c,\Omega}, y_{c,\Omega}) + \frac{1}{c}\|\lambda_c\|^2_{L^2(D)} \leq a(y_{c,\Omega}, y_\Omega) + (f_\Omega, y_{c,\Omega} - y_\Omega)_{L^2(D)} + \frac{1}{c}\left(\lambda_c, \overline{\lambda}\right)_{L^2(D)}.$$

Using $a(v,v) \geq \mu\|v\|^2_{H^1_0(D)}$ for all $v \in H^1_0(D)$ and some $\mu > 0$ due to coercivity, $\|v\|_{L^2(D)} \leq \|v\|_{H^1_0(D)}$ and boundedness of $a(.,.)$ with constant $\varrho > 0$ we get

$$\begin{aligned}\mu\|y_{c,\Omega}\|^2_{H^1_0(D)} + \frac{1}{c}\|\lambda_c\|^2_{L^2(D)} \leq\, & a(y_{c,\Omega}, y_\Omega) + (f_\Omega, y_{c,\Omega} - y_\Omega)_{L^2(D)} + \frac{1}{c}\left(\lambda_c, \overline{\lambda}\right)_{L^2(D)} \\ \leq\, & \varrho\|y_{c,\Omega}\|_{H^1_0(D)}\|y_\Omega\|_{H^1_0(D)} + \|f_\Omega\|_{L^2(D)}\|y_{c,\Omega}\|_{H^1_0(D)} \\ & + \|f_\Omega\|_{L^2(D)}\|y_\Omega\|_{H^1_0(D)} + \frac{1}{c}\|\lambda_c\|_{L^2(D)}\|\overline{\lambda}\|_{L^2(D)}.\end{aligned} \tag{3.14}$$

Young's inequality gives us for any $s, t, \varepsilon > 0$ the following inequality

$$st \leq \frac{s^2}{2\varepsilon} + \frac{\varepsilon t}{2}.$$

Thus we can estimate for three of the terms on the right-hand side of (3.14)

$$\begin{aligned}\frac{1}{c}\|\lambda_c\|_{L^2(D)}\|\overline{\lambda}\|_{L^2(D)} &\leq \frac{1}{2c}(\|\lambda_c\|^2_{L^2(D)} + \|\overline{\lambda}\|^2_{L^2(D)}), \\ \varrho\|y_{c,\Omega}\|_{H^1_0(D)}\|y_\Omega\|_{H^1_0(D)} &\leq \varrho\frac{\|y_\Omega\|^2_{H^1_0(D)}}{2\varepsilon} + \varrho\frac{\varepsilon\|y_{c,\Omega}\|^2_{H^1_0(D)}}{2}, \\ \varrho\|y_{c,\Omega}\|_{H^1_0(D)}\|f_\Omega\|_{L^2(D)} &\leq \varrho\frac{\|f_\Omega\|^2_{L^2(D)}}{2\varepsilon} + \varrho\frac{\varepsilon\|y_{c,\Omega}\|^2_{H^1_0(D)}}{2}.\end{aligned}$$

Using those inequalities we obtain for $\varepsilon < \mu/\varrho$ after rearranging some terms in (3.14)

$$\begin{aligned}\underbrace{(\mu - \varrho\varepsilon)}_{>0}\|y_{c,\Omega}\|_{H^1_0(D)} + \frac{1}{2c}\|\lambda_c\|^2_{L^2(D)} \leq\, & \varrho\frac{\|y_\Omega\|^2_{H^1_0(D)}}{2\varepsilon} + \|f_\Omega\|_{L^2(D)}\|y_\Omega\|_{H^1_0(D)} \\ & + \varrho\frac{\|f_\Omega\|^2_{L^2(D)}}{2\varepsilon} + \frac{1}{2c}\|\overline{\lambda}\|^2_{L^2(D)}.\end{aligned}$$

From this inequality we conclude that $y_{c,\Omega} \in H^1_0(D)$ must be bounded for $c \geq 1$ as this inequality could not hold true otherwise. Since λ_c solves (3.10) and $y_{c,\Omega}$ is bounded also λ_c has to be bounded in $H^{-1}(D)$. Therefore there exists a sub-sequence $(y_{c_n,\Omega}, \lambda_{c_n})_{n\in\mathbb{N}}$ of $(y_{c,\Omega}, \lambda_c)_{c\geq 1}$ and some $(\widetilde{y}_\Omega, \widetilde{\lambda}) \in H^1_0(D) \times H^{-1}(D)$ such that

$$(y_{c_n,\Omega}, \lambda_{c_n}) \rightharpoonup \left(\widetilde{y}_\Omega, \widetilde{\lambda}\right) \in H^1_0(D) \times H^{-1}(D),$$

i.e. $(y_{c_n,\Omega}, \lambda_{c_n})$ converges weakly.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Since $\|\lambda_{c_n}\|^2_{L^2(D)}$ is bounded, the left-hand side of

$$\frac{1}{c_n}\|\lambda_{c_n}\|^2_{L^2(D)} = c_n \left\| \max\left(0, \frac{\overline{\lambda}}{c_n} + y_{c_n,\Omega} - \varphi\right)\right\|^2_{L^2(D)}$$

converges to 0 as $c_n \to +\infty$. The max(...)-function is non-negative and measurable, therefore we can apply the lemma of Fatou. The sequence $y_{c_n,\Omega}$ converges to $\widetilde{y}_\Omega$ weakly in $H^1_0(D)$ and therefore strongly to $\widetilde{y}_\Omega$ in $L^2(D)$. Thus we obtain

$$\begin{aligned}
0 &\geq \liminf_{n\to+\infty} \int_D \max\left(0, \frac{\overline{\lambda}}{c_n} + y_{c_n,\Omega} - \varphi\right) \max\left(0, \frac{\overline{\lambda}}{c_n} + y_{c_n,\Omega} - \varphi\right) \, \mathrm{d}x \\
&\geq \int_D \liminf_{n\to+\infty} \max\left(0, \frac{\overline{\lambda}}{c_n} + y_{c_n,\Omega} - \varphi\right) \max\left(0, \frac{\overline{\lambda}}{c_n} + y_{c_n,\Omega} - \varphi\right) \, \mathrm{d}x \\
&= \int_D \max\left(0, \widetilde{y}_\Omega - \varphi\right) \max\left(0, \widetilde{y}_\Omega - \varphi\right) \, \mathrm{d}x \geq 0
\end{aligned}$$

and conclude that $\widetilde{y}_\Omega \leq \varphi$ almost everywhere in D.

From (3.13) we calculate futher

$$\begin{aligned}
(\lambda_c, y_{c,\Omega} - y_\Omega)_{L^2(D)} &\geq \frac{1}{c}\|\lambda_c\|^2_{L^2(D)} - \frac{1}{c}\left(\lambda_c, \overline{\lambda}\right)_{L^2(D)} \\
&= \frac{1}{c}\Big(\left(\lambda_c, \lambda_c\right)_{L^2(D)} - \left(\lambda_c, \overline{\lambda}\right)_{L^2(D)} \\
&\qquad + \left(\tfrac{1}{2}\overline{\lambda}, \tfrac{1}{2}\overline{\lambda}\right)_{L^2(D)} - \left(\tfrac{1}{2}\overline{\lambda}, \tfrac{1}{2}\overline{\lambda}\right)_{L^2(D)}\Big) \\
&= \frac{1}{c}\left\|\lambda_c - \tfrac{1}{2}\overline{\lambda}\right\|^2_{L^2(D)} - \frac{1}{4c}\left\|\overline{\lambda}\right\|^2_{L^2(D)} \\
&\geq -\frac{1}{4c}\left\|\overline{\lambda}\right\|^2_{L^2(D)}
\end{aligned}$$

and hence $\lim_{n\to+\infty} -(\lambda_{c_n}, y_{c_n,\Omega} - y_\Omega)_{L^2(D)} \leq 0$. As (3.10) and (3.4) both have the same right hand-side we get

$$a(y_{c_n,\Omega} - y_\Omega, y_{c_n,\Omega} - y_\Omega) + (\lambda_{c_n} - \lambda, y_{c_n,\Omega} - y_\Omega)_{L^2(D)} = 0. \tag{3.15}$$

Using coercivity of $a(.,.)$ and (3.15) we get

$$0 \leq \limsup_{n\to\infty} \mu\|y_{c_n,\Omega} - y_\Omega\|^2_{H^1_0(D)} \leq \limsup_{n\to+\infty} \langle \lambda - \lambda_c, y_{c_n,\Omega} - y_\Omega\rangle_{H^{-1}(D)\times H^1_0(D)}.$$

Using $\lim_{n\to+\infty} -(\lambda_{c_n}, y_{c_n,\Omega} - y_\Omega)_{L^2(D)} \leq 0$, $(\lambda, y_\Omega - \varphi)_{L^2(D)} = 0$ and $\widetilde{y}_\Omega \leq \varphi$ we can

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

further estimate

$$\begin{aligned}
0 \leq \limsup_{n\to\infty} \mu\|y_{c_n,\Omega} - y_\Omega\|^2_{H_0^1(D)} &\leq \limsup_{n\to+\infty} \langle \lambda - \lambda_{c_n}, y_{c_n,\Omega} - y_\Omega\rangle_{H^{-1}(D)\times H_0^1(D)} \\
&\leq \limsup_{n\to+\infty} \langle \lambda, y_{c_n,\Omega} - y_\Omega\rangle_{H^{-1}(D)\times H_0^1(D)} \\
&= \limsup_{n\to+\infty} \langle \lambda, y_{c_n,\Omega} - \varphi - (y_\Omega - \varphi)\rangle_{H^{-1}(D)\times H_0^1(D)} \\
&= \langle \lambda, \widetilde{y}_\Omega - \varphi\rangle_{H^{-1}(D)\times H_0^1(D)} \leq 0.
\end{aligned}$$

Since $0 \leq \liminf_{n\to+\infty} \mu\|y_{c_n,\Omega} - y_\Omega\|^2_{H_0^1(D)} \leq \limsup_{n\to+\infty} \mu\|y_{c_n,\Omega} - y_\Omega\|^2_{H_0^1(D)} \leq 0$ the limits coincide and we obtained $y_{c_n,\Omega} \to y_\Omega$ in $H_0^1(D)$ and consequently $\widetilde{y}_\Omega = y_\Omega$. To conclude the proof we notice that the limit of

$$a(y_{c_n,\Omega}, v) + (\lambda_{c_n}, v)_{L^2(D)} = (f_\Omega, v)_{L^2(D)} \quad \text{for all } v \in H_0^1(D),$$

is given by

$$a(y_\Omega, v) + \left\langle \widetilde{\lambda}, v \right\rangle_{H^{-1}(D)\times H_0^1(D)} = (f_\Omega, v)_{L^2(D)} \quad \text{for all } v \in H_0^1(D).$$

This last equality holds also true if we replace $\widetilde{\lambda}$ by λ and thus $\lambda_c \rightharpoonup \lambda$ weakly in $H^{-1}(D)$ as $c \to +\infty$. □

Theorem 3.6. *Let the assumptions of Theorem 3.5 hold. Let the function* $\max_\gamma$ *satisfy Assumption 3.2. Then* (3.11) *has a unique solution* $y_{\gamma,c,\Omega} \in H_0^1(D)$ *and*

$$y_{\gamma,c,\Omega} \to y_{c,\Omega} \quad \text{in } H^1(D) \quad \text{as } \gamma \to +\infty.$$

Proof. The existence of a unique solution can be shown in the same way as in the previous theorem. To show convergence we use the coercivity of $a(.,.)$ with constant $\mu > 0$ and monotonicity of max giving us

$$\begin{aligned}
0 &\leq \mu\|y_{\gamma,c,\Omega} - y_{c,\Omega}\|^2_{H_0^1(D)} \\
&\leq a(y_{\gamma,c,\Omega} - y_{c,\Omega}, y_{\gamma,c,\Omega} - y_{c,\Omega}) \\
&\leq a(y_{\gamma,c,\Omega} - y_{c,\Omega}, y_{\gamma,c,\Omega} - y_{c,\Omega}) \\
&\quad + \left(\max\left(0, \overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) - \max\left(0, \overline{\lambda} + c(y_{c,\Omega} - \varphi)\right), y_{\gamma,c,\Omega} - y_{c,\Omega}\right)_{L^2(D)}.
\end{aligned} \tag{3.16}$$

As $y_{\gamma,c,\Omega} - y_{c,\Omega} \in H_0^1(D)$ and since both $y_{\gamma,c,\Omega}$ and $y_{c,\Omega}$ are solutions of their respective state equations we have

$$\begin{aligned}
&- a(y_{c,\Omega}, y_{\gamma,c,\Omega} - y_{c,\Omega}) - \left(\max\left(0, \overline{\lambda} + c(y_{c,\Omega} - \varphi)\right), y_{\gamma,c,\Omega} - y_{c,\Omega}\right)_{L^2(D)} \\
&= -(f_\Omega, y_{\gamma,c,\Omega} - y_{c,\Omega})_{L^2(D)} \\
&= -a(y_{\gamma,c,\Omega}, y_{\gamma,c,\Omega} - y_{c,\Omega}) - \left(\max_\gamma\left(0, \overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right), y_{\gamma,c,\Omega} - y_{c,\Omega}\right)_{L^2(D)}.
\end{aligned} \tag{3.17}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Plugging this equation into (3.16), using Assumption 3.2 *(ii)*, the Cauchy-Schwarz inequality and $\|y_{\gamma,c,\Omega}-y_{c,\Omega}\|_{L^2(D)} \leq \|y_{\gamma,c,\Omega}-y_{c,\Omega}\|_{H_0^1(D)}$ we can further estimate (3.17)

$$\begin{aligned}
&= \left(\max\left(0,\overline{\lambda}+c(y_{\gamma,c,\Omega}-\varphi)\right)-\max_\gamma\left(0,\overline{\lambda}+c(y_{c,\Omega}-\varphi)\right), y_{\gamma,c,\Omega}-y_{c,\Omega}\right)_{L^2(D)} \\
&\leq \int_D \left|\max\left(0,\overline{\lambda}+c(y_{\gamma,c,\Omega}-\varphi)\right)-\max_\gamma\left(0,\overline{\lambda}+c(y_{\gamma,c,\Omega}-\varphi)\right)\right| |y_{\gamma,c,\Omega}-y_{c,\Omega}| \ \mathrm{d}x \\
&\leq g(\gamma)\mathrm{vol}(D)^{1/2}\|y_{\gamma,c,\Omega}-y_{c,\Omega}\|_{H_0^1(D)},
\end{aligned}$$

where $\mathrm{vol}(D)$ denotes the volume of D, which is finite by assumption. Dividing the inequality by $\|y_{\gamma,c,\Omega}-y_{c,\Omega}\|_{H_0^1(D)}$ shows convergence. □

Remark 3.7. In [17] the convergences of $p_{\gamma,c,\Omega}$ to some $p_{c,\Omega}$ and of $p_{c,\Omega}$ to some p_Ω in $H_0^1(D)$ as $\gamma \to +\infty$ and $c \to +\infty$ respectively are shown. The authors are able to show this result using the following additional assumption

$$\left\|\max_\gamma'\left(\overline{\lambda}+c(y_{\gamma,c,\Omega}-\varphi)\right)-\max'\left(0,\overline{\lambda}+c(y_{c,\Omega}-\varphi)\right)\right\|_{L^1(D)} \to 0 \quad \text{as} \quad \gamma \to +\infty.$$

This assumption is not trivial and the authors are aware of this fact. In their numerical experiments they check numerically if this condition holds true for their chosen $\max_\gamma$. Since no motivation is given for why this convergence should hold true in an infinite-dimensional setting we prove a different, weaker result. Our result is more in line with similar results in the literature, see for example [4, Proposition 2.1]. We only show that a subsequence of $p_{\gamma,c,\Omega}$ converges weakly to some $p_{c,\Omega}$ in $H_0^1(D)$ as $\gamma \to +\infty$. It is an open question if the whole sequence $p_{\gamma,c,\Omega}$ converges to $p_{c,\Omega}$, if $p_{c,\Omega}$ converges in some sense to some p_Ω and how this p_Ω relates to the unregularized problem. Note that the unregularized problem has no adjoint equation as it lacks regularity.

Theorem 3.8. *Let $D \subset \mathbb{R}^d$ be an open, bounded domain with Lipschitz boundary ∂D, the bilinear form $a(.,.)$ be as in Section 3.1, $f_\Omega \in L^2(D)$, $c > 0$, $\varphi \in H^1(D)$ and $\overline{\lambda} \in L^2(D)$, let y_Ω be the solution to* (3.4)*. Let $\max_\gamma$ satisfy Assumption 3.2 and let $y_{c,\Omega} \in H_0^1(D)$ solve* (3.10)*.*

Then (3.12) *has a unique solution $p_{\gamma,c,\Omega} \in H_0^1(D)$ and there exists a subsequence $(p_{\gamma_n,c,\Omega})_{n\in\mathbb{N}}$ of $p_{\gamma,c,\Omega}$ such that $p_{\gamma_n,c,\Omega} \rightharpoonup p_{c,\Omega}$ weakly in $H_0^1(D)$ for all $c > 0$ as $n \to +\infty$. Here $p_{c,\Omega}$ is the solution to*

$$a(p_{c,\Omega},v)+c\int_D \mathbb{1}_A p_{c,\Omega} v \ \mathrm{d}x+\int_D \mathbb{1}_{A_0} z v \ dx = -\int_D (y_{c,\Omega}-\overline{y})v \ \mathrm{d}x \quad \forall v \in H_0^1(D) \quad (3.18)$$

for some $z \in L^\infty(D)'$ and the sets A and A_0 given by $A := \{x \in \Omega : \overline{\lambda}+c(y_{c,\Omega}-\varphi) > 0\}$ and $A_0 := \left\{x \in D : \overline{\lambda}+c(y_{c,\Omega}-\varphi) = 0\right\}$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. Existence and uniqueness of the solution $p_{\gamma,c,\Omega}$ of the fully regularized adjoint equation follow from the Riesz representation theorem since the operator $\mathbf{S}_{\gamma,c}$ defined by

$$\begin{aligned}\mathbf{S}_{\gamma,c} : H_0^1(D) &\to H^{-1}(D)\\ p &\mapsto a(p,\ .\ ) + c\left(\max_\gamma'\left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p,\ .\ \right)_{L^2(D)}\end{aligned}$$

satisfies the conditions of the Lax-Milgram Lemma. Note that coercivity of $\mathbf{S}_{\gamma,c}$ follows from the coercivity of $a(.,.)$ and $\max_\gamma'\left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) \geq 0$ due to Assumption 3.2 (iii).

We now want to show that (3.12) converges to (3.18) as $\gamma \to +\infty$. Thus we want to calculate the limit

$$\begin{aligned}\lim_{\gamma\to+\infty} a(p_{\gamma,c,\Omega}, v) + c(\max_\gamma'(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))p_{\gamma,c,\Omega}, v)_{L^2(D)}&\\ = \lim_{\gamma\to+\infty} -(y_{\gamma,c,\Omega} - \overline{y}, v)_{L^2(D)}&\end{aligned} \tag{3.19}$$

for $v \in H_0^1(D)$. Since $y_{\gamma,c,\Omega} \to y_{c,\Omega}$ strongly in $H_0^1(D)$ as $\gamma \to +\infty$ the right-hand side converges to

$$-(y_{c,\Omega} - \overline{y}, v)_{L^2(D)}.$$

Next we show that $p_{\gamma,c,\Omega}$ is bounded in $H_0^1(D)$. Let $\|p_{\gamma,c,\Omega}\|_{H_0^1(D)} \neq 0$. Due to coercivity of $a(.,.)$ with constant $\mu > 0$, $\max_\gamma'(x) \in [0,1]$ by Assumption 3.2 (iii) and $p_{\gamma,c,\Omega}$ solving (3.12) we have

$$\begin{aligned}\mu\|p_{\gamma,c,\Omega}\|^2_{H_0^1(D)} &\leq a(p_{\gamma,c,\Omega}, p_{\gamma,c,\Omega})\\ &\leq a(p_{\gamma,c,\Omega}, p_{\gamma,c,\Omega}) + c(\max_\gamma'(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))p_{\gamma,c,\Omega}, p_{\gamma,c,\Omega})_{L^2(D)}\\ &= -(y_{c,\Omega} - \overline{y}, p_{\gamma,c,\Omega})_{L^2(D)}\\ &\leq \|y_{c,\Omega} - \overline{y}\|_{L^2(D)}\|p_{\gamma,c,\Omega}\|_{H_0^1(D)}.\end{aligned}$$

Dividing both sides by $\|p_{\gamma,c,\Omega}\|_{H_0^1(D)}$ yields us boundedness of $p_{\gamma,c,\Omega}$ in $H_0^1(D)$.

Consequently there exists a sub-sequence $(p_{\gamma_n,c,\Omega})_{n\in\mathbb{N}}$ and some $p_{c,\Omega} \in H_0^1(D)$ such that $p_{\gamma,c,\Omega} \rightharpoonup p_{c,\Omega}$ weakly in $H_0^1(D)$ as $n \to +\infty$. We now restrict the limit in (3.19) to the limit of this sub-sequence.

For the term containing the $\max_\gamma'$ we split our domain D into the three parts A, A_0 and $D\backslash(A \cup A_0)$. For A we have to calculate the limit of

$$\lim_{n\to+\infty} c(\max_{\gamma_n}'(\overline{\lambda} + c(y_{\gamma_n,c,\Omega} - \varphi))p_{\gamma_n,c,\Omega}, v)_{L^2(A)}.$$

Since again $y_{\gamma_n,c,\Omega} \to y_{c,\Omega}$ strongly in $H_0^1(D)$ there exists a $\gamma^* > 0$ such that A and $\{x \in \Omega : \overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi) > 0\}$ only differ on a set of measure 0 for $\gamma \geq \gamma^*$. Applying

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Lebesuge's dominated convergence theorem and $\lim_{n\to+\infty} \max'_{\gamma_n}(x) = 1$ for $x \in A$ we obtain

$$\lim_{n\to+\infty} c(\max'_{\gamma_n}(\overline{\lambda} + c(y_{\gamma_n,c,\Omega} - \varphi))p_{\gamma_n,c,\Omega}, v)_{L^2(A)} = c\int_D \mathbb{1}_A p_{c,\Omega} v \,\mathrm{d}x.$$

The reasoning for the set $D\backslash(A \cup A_0)$ is the same with the difference that the limit of $\max'_{\gamma_n}$ is 0 there. Therefore it vanishes.

It remains to calculate

$$\lim_{n\to+\infty} c(\max'_{\gamma_n}(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))p_{\gamma_n,c,\Omega}, v)_{L^2(A_0)}.$$

Since $\max'_\gamma \in [0,1]$ due to Assumption 3.2 (iv) for all $\gamma > 0$ the sequence $(\max'_\gamma)_\gamma$ is bounded. Therefore there exists a subsequence $(\max'_{\gamma_k})_{k\in\mathbb{N}}$ of the sequence $(\max'_{\gamma_n})_{n\in\mathbb{N}}$ such that $\max'_{\gamma_k} \rightharpoonup z \in L^\infty(A_0)'$. From this we obtain convergence of

$$c\int_{A_0} \max'_{\gamma_k}(\overline{\lambda} + c(y_{\gamma_k,c,\Omega} - \varphi))p_{\gamma_k,c,\Omega} v \,\mathrm{d}x \to c\int_{A_0} z p_{c,\Omega} v \,\mathrm{d}x$$

or put differently: $\max'_{\gamma_k}(\overline{\lambda} + c(y_{\gamma_k,c,\Omega} - \varphi))p_{\gamma_k,c,\Omega} v$ converges in $L^1(A_0)$. Thus

$$\lim_{k\to+\infty} c(\max'_{\gamma_k}(\overline{\lambda} + c(y_{\gamma_k,c,\Omega} - \varphi))p_{\gamma_k,c,\Omega}, v)_{L^2(A_0)} = c(z p_{c,\Omega}, v)_{L^2(A_0)}$$

for all $v \in L^2(D)$. This yields (3.18). □

Remark 3.9. In [17, Theorem 3.3] the authors obtain the same formula for $p_{c,\Omega}$ but without our additional term $\int_D \mathbb{1}_{A_0} z v \,\mathrm{d}x$.

3.4. A Shape Derivative Formula

In this section we want to show the existence of the shape derivative $\mathrm{d}J(\Omega)$ of our problem and give an explicit formula. We start by giving a short motivation for Lagrange functions before we introduce the averaged adjoint approach which we then use to show the existence of a shape derivative for our fully regularized problem. We calculate the explicit formula of the shape derivative in a more direct way using fewer assumptions than in [17, Theorem 3.4].

3.4.1. On Lagrange Functions

Closely connected to our minimization problem is a certain Lagrange function $\mathcal{L}$. Let us recall from the finite dimensional setting the notion of Lagrange multipliers.

Given some continuously differentiable functions $h : D \to \mathbb{R}$, with $D \subset \mathbb{R}^d$ open and $g : D \to \mathbb{R}^{d-\ell}$ we want to find local extrema of $h|_M : M \to \mathbb{R}$ where M is a manifold defined implicitly by $M := \{x \in D : g(x) = 0\}$. In this setting we have the Lagrange multiplier theorem.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Theorem 3.10 (Lagrange multiplier)**.** *Given the above setting, if the function $h : D \to \mathbb{R}$ has a local extremum at $y \in D$ satisfying $g(y) = 0$ and $\mathrm{d}g(y)$ has full rank $d - \ell$, then there exists a $\mu \in \mathbb{R}^{d-\ell}$, such that for the function*

$$\begin{aligned} \mathcal{L}_{\text{fin}} : D \times \mathbb{R}^{d-\ell} &\to \mathbb{R} \\ (x, \lambda) &\mapsto h(x) + \sum_{i=1}^{d-\ell} \lambda_i g_i(x) \end{aligned}$$

(y, μ) is a stationary point, i.e. it satisfies $\mathrm{d}\mathcal{L}_{\text{fin}}(y, \mu) = 0$. The number λ is called the Lagrange multiplier*.*

A proof can be found in [14, Satz 14.8.1]. The aim of Lagrange multipliers is to shift the problem to one that is easier. Instead of finding a stationary point y of h on some implicitly defined set M we try to find stationary points $(\widetilde{y}, \widetilde{\mu})$ of $\mathcal{L}_{\text{fin}}$ on the whole domain. Every such found $\widetilde{y}$ then is a candidate of also being a stationary point of h and fulfilling $g(\widetilde{y}) = 0$.

This idea can be generalized to Banach spaces. Let us first define our Lagrangian function $\mathcal{L}$ and then point out the similarities to the finite dimensional case. Denote by $\mathcal{A}$ the set of admissible domains. We define

$$\begin{aligned} \mathcal{L} : \mathcal{A} \times H_0^1(D) \times H_0^1(D) &\to \mathbb{R} \\ (\Omega, y, p) &\mapsto \frac{1}{2} \int_D (y - \overline{y})^2 \, \mathrm{d}x + a(y, p) + \big(\max_\gamma(\overline{\lambda} + c(y - \varphi)), p\big)_{L^2(D)} - (f_\Omega, p)_{L^2(D)} \,. \end{aligned} \tag{3.20}$$

We assume Ω to be fixed for now. The first term $1/2 \int_D (y - \overline{y})^2 \, \mathrm{d}x$ is the objective function we want to minimize. The other three terms

$$a(y, p) + \big(\max_\gamma(\overline{\lambda} + c(y - \varphi)), p\big)_{L^2(D)} - (f_\Omega, p)_{L^2(D)}$$

are our fully regularized state equation (3.11) with the right hand-side brought to the left. Therefore we have as in the finite dimensional case a form $g(x) = 0$.

The function $p \in H_0^1(D)$ takes the role of the Lagrange multiplier. If we differentiate the finite dimensional Lagrange function $\mathcal{L}_{\text{fin}}$ by λ in any direction $v \in \mathbb{R}^{d-\ell}$ we obtain $g(x)$. We are only interested in $x \in D$ for which $g(x) = 0$ or equivalently $\sum_{i=1}^{d-\ell} v_i g_i(x) = 0$. In the infinite dimensional setting if we differentiate $\mathcal{L}$ by p in any direction v we get our state equation back. Here we are only interested in $y \in H_0^1(D)$ which satisfy

$$a(y, v) + \big(\max_\gamma(\overline{\lambda} + c(y - \varphi)), v\big)_{L^2(D)} - (f_\Omega, v)_{L^2(D)} = 0$$

for all $v \in H_0^1(D)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

We can now also finally see where the adjoint equation stems from. In the finite dimensional setting we look for points $(\widetilde{y}, \widetilde{\mu})$ of $\mathcal{L}_{\text{fin}}$ which satisfy

$$\mathrm{d}\mathcal{L}_{\text{fin}}(\widetilde{y}, \widetilde{\mu}) = \frac{\partial \mathcal{L}_{\text{fin}}}{\partial x}(\widetilde{y}, \widetilde{\mu}) + \frac{\partial \mathcal{L}_{\text{fin}}}{\partial \lambda}(\widetilde{y}, \widetilde{\mu}) = 0.$$

Since $\partial_\lambda \mathcal{L}_{\text{fin}}(\widetilde{y}, \widetilde{\mu}) = 0$ if $\widetilde{y}$ satisfies $g(\widetilde{y}) = 0$ we conclude that only points which also satisfy $\partial_x \mathcal{L}_{\text{fin}}(\widetilde{y}, \widetilde{\mu}) = 0$ are possible candidates for a solution to our problem.

We require the same condition in the infinite dimensional setting, i.e.

$$\frac{\partial \mathcal{L}}{\partial y}(\widetilde{y}, \widetilde{p}) + \frac{\partial \mathcal{L}}{\partial p}(\widetilde{y}, \widetilde{p}) = 0$$

for some $(\widetilde{y}, \widetilde{p}) \in H_0^1(D) \times H_0^1(D)$. Remember that we still assume Ω to be fixed. Forcing the first term in this equation to be 0 as well yields our adjoint equation (3.12).

Finally to see the connection between our Lagrange function $\mathcal{L}$ and our shape function $J(\Omega)$ we note that for the finite dimensional case we have

$$\sup_{\lambda \in \mathbb{R}^{d-\ell}} \mathcal{L}_{\text{fin}}(x, \lambda) = \begin{cases} h(x) & \text{if } x \text{ solves } g(x) = 0, \\ +\infty & \text{else.} \end{cases}$$

Similar in the infinte dimensional case we have

$$\sup_{p \in H_0^1(D)} \mathcal{L}(\Omega, y, p) = \begin{cases} J(\Omega) & \text{if } y \text{ solves (3.11)} \\ +\infty & \text{else} \end{cases}$$

and thus

$$J(\Omega) = \min_{y \in H_0^1(D)} \sup_{p \in H_0^1(D)} \mathcal{L}(\Omega, y, p).$$

3.4.2. The Averaged Adjoint Approach

The average adjoint approach was first introduced in [20]. This approach introduces a slightly modified adjoint equation, the so called *averaged adjoint equation*. Averaged in this case means taking the integral over the adjoint equation in a special way. We show that our problem satisfies all assumptions of Theorem 3.11, the main theorem in this section. It gives us existence of the shape derivative $\mathrm{d}J(\Omega)$. We then go on and calculate an explicit formula for $\mathrm{d}J(\Omega)$.

Let $\theta \in C_c^1(\overline{D}, \mathbb{R}^d)$ be given. We define

$$G : [0, \tau] \times H_0^1(D) \times H_0^1(D) \to \mathbb{R}, \ (t, y, p) \mapsto \mathcal{L}(\Omega_t, y \circ (\Phi_t^\theta)^{-1}, p \circ (\Phi_t^\theta)^{-1}), \quad (3.21)$$

where $\mathcal{L}$ is defined as in (3.20). Due to [21, Theorem 7.1] we have the following theorem.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Theorem 3.11. *Let E and F be normed vector spaces and let $\tau > 0$. Let the function*

$$G : [0,\tau] \times E \times F : (t,y,p) \mapsto G(t,y,p)$$

be affine linear in the last argument. We define $y^t := y \circ \Phi_t^\theta$,

$$E(t) := \{y \in E : \mathrm{d}_p G(t,y,0;\widehat{p}) = 0 \text{ for all } \widehat{p} \in F\}$$

and

$$Y(t,y^t,y^0) := \left\{p \in F : \text{For all } \widehat{y} \in E : \int_0^1 \mathrm{d}_y G(t, sy^t + (1-s)y^0, p; \widehat{y})\mathrm{d}s = 0\right\}.$$

Let the following conditions hold:

(H0) *(i)* *$E(t)$ is single-valued for all $t \in [0,\tau]$.*

(ii) *For all $t \in [0,\tau]$, $y^t \in E(t)$, $y^0 \in E(0)$ and $p \in F$ the mapping*

$$[0,1] \to \mathbb{R} : s \mapsto G(t, sy^t + (1-s)y^0, p)$$

is absolutely continuous.

(iii) *For all $t \in [0,\tau]$, $y^t \in E(t)$, $y^0 \in E(0)$ and for all $\widehat{y} \in E$ and $p \in F$ the mapping*

$$s \mapsto \mathrm{d}_y G(t, sy^t + (1-s)y^0, p; \widehat{y})$$

is well defined and belongs to $L^1(0,1)$.

(H1) *For all $t \in [0,\tau]$ and all $(y^0,p) \in E(0) \times F$ the derivative $\partial_t G(t,y^0,p)$ exists.*

(H2) *For all $t \in [0,\tau]$ the set $Y(t,y^t,y^0)$ is non-empty and $Y(0,y^0,y^0)$ is single-valued.*

(H3) *Let $p^0 \in Y(0,y^0,y^0)$. For any sequence $(t_n)_{n\in\mathbb{N}}$ of non-negative real numbers converging to 0 there exists a subsequence $(t_{n_k})_{k\in\mathbb{N}}$ and $p^{t_{n_k}} \in Y(t_{n_k}, y^{t_{n_k}}, y^0)$ such that*

$$\lim_{\substack{k\to+\infty \\ s\to 0^+}} \partial_t G(s, y^0, p^{t_{n_k}}) = \partial_t G(0,y^0,p^0).$$

Then it holds for any $p \in F$

$$\frac{\mathrm{d}}{\mathrm{d}t} G(t,y^t,p)|_{t=0} = \partial_t G(0,y^0,p^0) = \mathrm{d}J(\Omega)(\theta). \tag{3.22}$$

In our setting of course E and F are both $H_0^1(D)$. The set $E(t)$ is the solution set of our state equation (3.11) with f_{Ω_t} instead of f_Ω. The set $Y(t,y^t,y^0)$ is called solution set of the averaged adjoint equation. Assumption *H(0) (ii)* implies by definition of absolute continuity that for almost all $s \in [0,1]$ the derivative $\mathrm{d}_y G(t, sy^t + (1-s)y^0, p; y^t - y^0)$ exists.

In order to calculate the shape derivative $\mathrm{d}J_{\gamma,c}$ we need the so called material derivative.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Definition 3.12. Let $D \subset \mathbb{R}^d$ be bounded and open and $\Omega \subset D$. Let $\theta \in C_c^1(D, \mathbb{R}^d)$, $\tau > 0$ and $(\Phi_t^\theta)_{t\in[0,\tau]}$ be the flows of θ. Let $\Omega_t := \Phi_t^\theta(\Omega) = \{\Phi_t^\theta(x) \in D : x \in \Omega\}$. Let $(p_{\Omega_t})_{t\in[0,\tau]}$ be a family of mappings with $p_{\Omega_t} : D \to \mathbb{R}$ depending on the shape Ω_t.

The *material derivative* $\dot{p}_\theta$ of a function $p_\Omega : D \to \mathbb{R}$ at $x \in D$ is given by

$$\dot{p}_\theta(x) := \lim_{t\to 0^+} \frac{(p_{\Omega_t} \circ \Phi_t^\theta)(x) - p_\Omega(x)}{t}.$$

We now restate the theorem and proof on existence of the shape derivative for the fully regularized problem as found in [17, Appendix A].

Theorem 3.13. *Let $D \subset \mathbb{R}^d$ be an open, bounded domain with Lipschitz boundary ∂D, the bilinear form $a(.,.)$ be as in Section 3.1 with $(m_{i,j})_{i,j}$, $(e_i)_i$ and b in $L^\infty(D) \cap H^1(D)$. Let the material derivative $\dot{f}_\theta$ of $f_\Omega \in L^2(D)$ exist. Let $c > 0$, $\varphi \in H^1(D)$ and $\overline{\lambda} \in H^1(D)$. Let y_Ω be the solution to (3.4). Let $\theta \in C_c^1(D, \mathbb{R}^d)$ be a vector field.*

Then the shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ exists.

Proof. We have to verifiy the assumptions *(H0)-(H3)* of Theorem 3.11 for the function G as defined in (3.21). If we write out the function G we have

$$\begin{aligned} G(t,y,p) = \frac{1}{2}\int_D (y \circ (\Phi_t^\theta)^{-1} - \overline{y})^2 \,\mathrm{d}x \\ + a(y \circ (\Phi_t^\theta)^{-1}, p \circ (\Phi_t^\theta)^{-1}) \\ + \left(\max_\gamma(\overline{\lambda} + c(y \circ (\Phi_t^\theta)^{-1} - \varphi)), p \circ (\Phi_t^\theta)^{-1}\right)_{L^2(D)} \\ - \left(f_{\Omega_t}, p \circ (\Phi_t^\theta)^{-1}\right)_{L^2(D)}. \end{aligned} \tag{3.23}$$

Since $a(.,.)$ and the scalar product are both affine linear we obtain that G is affine linear in the last argument.

Let $\tau > 0$ be small enough such that our transformation $\Phi_t^\theta : D \to D$ and $\Phi_t^\theta(\Omega) = \Omega_t$ is bijective.

Ad (H0) (i). In our specific setting we have

$$E(t) := \left\{y \in H_0^1(D) : \mathrm{d}_p G(t,y,0;v) = 0 \text{ for all } v \in H_0^1(D)\right\}.$$

In Section 3.4.1 we saw that $\mathrm{d}_p G(t,y,0;v) = 0$ for all $v \in H_0^1(D)$ is our fully regularized state equation. So this is just another way to say $E(t)$ is the set of solutions of the fully regularized state equation (3.11) with f_{Ω_t} instead of f_Ω. As proved in Theorem 3.6 this equation has a unique solution y_{γ,c,Ω_t} for every $t \in [0,\tau]$ and thus $E(t) = \{y^t_{\gamma,c,\Omega_t}\}$ with $y^t_{\gamma,c,\Omega_t} := y_{\gamma,c,\Omega_t} \circ \Phi_t^\theta$ is single-valued.

Ad (H0) (ii). To check absolute continuity of the function $[0,1] \to \mathbb{R} : s \mapsto G(t, sy^t_{\gamma,c,\Omega_t} + (1-s)y^0_{\gamma,c,\Omega}, p)$ we have to verify that this function has a Lebesgue

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

integrable derivative almost everywhere in $[0,1]$. The derivative is given by

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}s}G(t,sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega},p)\\
&\quad=\frac{\mathrm{d}}{\mathrm{d}s}\Big(\frac{1}{2}\int_D((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})\circ(\Phi^\theta_t)^{-1}-\overline{y})^2\,\mathrm{d}x\\
&\qquad+a((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})\circ(\Phi^\theta_t)^{-1},p\circ(\Phi^\theta_t)^{-1})\\
&\qquad+\big(\max\nolimits_\gamma(\overline{\lambda}+c((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})\circ(\Phi^\theta_t)^{-1}-\varphi)),p\circ(\Phi^\theta_t)^{-1}\big)_{L^2(D)}\\
&\qquad-\big(f_{\Omega_t},p\circ(\Phi^\theta_t)^{-1}\big)_{L^2(D)}\Big).
\end{aligned}$$

Using integration by substitution and $\Phi^\theta_t(D)=D$ we further calculate

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}s}G(t,sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega},p)\\
&\quad=\frac{\mathrm{d}}{\mathrm{d}s}\Big(\frac{1}{2}\int_D((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})-\overline{y}\circ\Phi^\theta_t)^2\det(\partial\Phi_t)\,\mathrm{d}x\\
&\qquad+a_t((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega}),p)\\
&\qquad+\big(\max\nolimits_\gamma(\overline{\lambda}\circ\Phi^\theta_t+c((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})-\varphi\circ\Phi^\theta_t)),p\det(\partial\Phi^\theta_t)\big)_{L^2(D)}\\
&\qquad-\big(f_{\Omega_t}\circ\Phi^\theta_t,p\det(\partial\Phi^\theta_t)\big)_{L^2(D)}\Big),
\end{aligned}$$

where $a_t(.,.)$ is the bilinear form obtained by integration by substitution to the unperturbed domain and is given by

$$\begin{aligned}
a_t(y,p):=\int_D&\big((\partial\Phi^\theta_t)^{-1}\nabla y^T(M\circ\Phi^\theta_t)(\partial\Phi^\theta_t)^{-1}\nabla p\\
&+(e\circ\Phi^\theta_t)^T(p(\partial\Phi^\theta_t)^{-1}\nabla y+y(\partial\Phi^\theta_t)^{-1}\nabla p)\\
&+(b\circ\Phi^\theta_t)yp\big)\det(\partial\Phi^\theta_t)\,\mathrm{d}x.
\end{aligned}\tag{3.24}$$

Finally differentiating this expression we get

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}s}G(t,sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega},p)\\
&\quad=\int_D((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})-\overline{y}\circ\Phi^\theta_t)(y^t_{\gamma,c,\Omega_t}-y^0_{\gamma,c,\Omega})\det(\partial\Phi^\theta_t)\,\mathrm{d}x\\
&\qquad+a_t((y^t_{\gamma,c,\Omega_t}-y^0_{\gamma,c,\Omega}),p)\\
&\qquad+\big(\max\nolimits'_\gamma(\overline{\lambda}\circ\Phi^\theta_t+c((sy^t_{\gamma,c,\Omega_t}+(1-s)y^0_{\gamma,c,\Omega})-\varphi\circ\Phi^\theta_t))(y^t_{\gamma,c,\Omega_t}-y^0_{\gamma,c,\Omega}),\\
&\qquad\qquad p\det(\partial\Phi^\theta_t)\big)_{L^2(D)},
\end{aligned}$$

which as a function of s is in $L^1([0,1])$, since $\max'_\gamma\in[0,1]$ due to Assumption 3.2 *(iii)* and all other terms are bounded.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Ad (H0) (iii). We have to show that for all $p, v \in H_0^1(D)$ the function

$$[0,1] \to \mathbb{R} : s \mapsto d_y G(t, s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}, p; v)$$

is integrable.

Again using integration by substitution to the unperturbed domain we have

$$\begin{aligned}
&\frac{\partial}{\partial y} G(t, s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}, p; v) \\
&\quad = \int_D \left((s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}) \circ (\Phi_t^\theta)^{-1} - \overline{y} \right) (v \circ (\Phi_t^\theta)^{-1}) \, \mathrm{d}x \\
&\qquad + a(v \circ (\Phi_t^\theta)^{-1}, p \circ (\Phi_t^\theta)^{-1}) \\
&\qquad + \int_D \max_\gamma'(\overline{\lambda} + c((s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}) \circ (\Phi_t^\theta)^{-1} - \varphi)) \\
&\qquad\qquad\qquad\qquad\qquad\qquad (v \circ (\Phi_t^\theta)^{-1})(p \circ (\Phi_t^\theta)^{-1}) \, \mathrm{d}x \\
&\quad = \left(s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - (\overline{y} \circ \Phi_t^\theta), v \det(\partial \Phi_t^\theta) \right)_{L^2(D)} \\
&\qquad + a_t(v, p) \\
&\qquad + \left(c \max_\gamma'(\overline{\lambda} \circ \Phi_t^\theta + c(s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - \varphi \circ \Phi_t^\theta)) v, \right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left. p \det(\partial \Phi_t^\theta) \right)_{L^2(D)} .
\end{aligned} \tag{3.25}$$

Since $\max_\gamma'(x) \in [0,1]$ for all $x \in \mathbb{R}$ and $\gamma > 0$) due to Assumption 3.2 *(iii)* and boundedeness of a_t with constant ϱ_t we get

$$\begin{aligned}
&\left\| \frac{\partial}{\partial y} G(t, s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}, \widetilde{p}; v) \right\|_{L^1(0,1)} \\
&\quad \le \int_0^1 \Big(\| s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - \overline{y} \circ \Phi_t^\theta)) \|_{L^2(D)} \| v \det(\partial \Phi_t^\theta) \|_{L^2(D)} \\
&\qquad\qquad + \varrho_t \| v \|_{H_0^1(D)} \| p \|_{H_0^1(D)} \\
&\qquad\qquad + c \| v \|_{L^2(D)} \| p \det(\partial \Phi_t^\theta) \|_{L^2(D)} \Big) \, \mathrm{d}s \\
&\quad \le \| \det(\partial \Phi_t^\theta) \|_\infty \Big(\int_0^1 \big(s \| y^t_{\gamma,c,\Omega_t} \|_{L^2(D)} + (1-s) \| y^0_{\gamma,c,\Omega} \|_{L^2(D)} \\
&\qquad\qquad\qquad + \| \overline{y} \circ \Phi_t^\theta \|_{L^2(D)} \big) \| v \|_{L^2(D)} \, \mathrm{d}s \\
&\qquad\qquad + (\widetilde{\varrho}_t + c) \| v \|_{H_0^1(D)} \| p \|_{H_0^1(D)} \Big) \\
&\quad = \| \det(\partial \Phi_t^\theta) \|_\infty \Big(\frac{1}{2} \big(\| y^t_{\gamma,c,\Omega_t} \|_{L^2(D)} + \| y^0_{\gamma,c,\Omega} \|_{L^2(D)} + \| \overline{y} \circ \Phi_t^\theta \|_{L^2(D)} \big) \| v \|_{L^2(D)} \\
&\qquad\qquad + (\widetilde{\varrho}_t + c) \| v \|_{H_0^1(D)} \| p \|_{H_0^1(D)} \Big) \\
&\quad < +\infty,
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

where $\widetilde{\varrho}_t$ is the boundedness constant of (3.24) with the term $\det(\partial\Phi_t^\theta)$ estimated from above by $\|\det(\partial\Phi_t^\theta)\|_\infty$.

Ad (H1). We explicitly calculate the derivative $\partial_t G(t, y^0_{\gamma,c,\Omega}, p)$ in the proof of Theorem 3.14.

Ad (H2). In our specific case the averaged adjoint set $Y(t, y^t, y^0)$ is defined as

$$Y(t, y^t_{\gamma,c,\Omega_t}, y^0_{\gamma,c,\Omega}) := \Big\{ p \in H_0^1(D) : \forall v \in H_0^1(D) : \int_0^1 \mathrm{d}_y G\left(t, s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}, p; v\right) \, \mathrm{d}s = 0 \Big\}.$$

As calculated in (3.25) we have

$$\begin{aligned} 0 &= \int_0^1 \mathrm{d}_y G\left(t, s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega}, p; v\right) \, \mathrm{d}s \\ &= \left(\frac{1}{2} y^t_{\gamma,c,\Omega_t} + \frac{1}{2} y^0_{\gamma,c,\Omega} - (\overline{y} \circ \Phi_t^\theta), v \det(\partial\Phi_t^\theta) \right)_{L^2(D)} \\ &\quad + a_t(v, p) \\ &\quad + \Bigg(c \int_0^1 \max_\gamma'(\overline{\lambda} \circ \Phi_t^\theta + c(s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - \varphi \circ \Phi_t^\theta)) v \, \mathrm{d}s, \\ &\qquad\qquad p \det(\partial\Phi_t^\theta) \Bigg)_{L^2(D)}. \end{aligned} \tag{3.26}$$

Again due to Assumption 3.2 *(iii)* we have some constant $c_t \in [0,1]$ only depending on t where c_t is defined as

$$c_t := \int_0^1 \max_\gamma'(\overline{\lambda} \circ \Phi_t^\theta + c(s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - \varphi \circ \Phi_t^\theta)) \, \mathrm{d}s.$$

So our set $Y(t, y^t_{\gamma,c,\Omega_t}, y^0_{\gamma,c,\Omega})$ consists of solutions of the problem: Find $q^t \in H_0^1(D)$ such that for all $v \in H_0^1(D)$ it holds

$$\begin{aligned} a_t(q^t, v) + c\left(c_t q^t, v \det(\partial\Phi_t^\theta)\right)_{L^2(D)} \\ = -\left(\frac{1}{2} y^t_{\gamma,c,\Omega_t} + \frac{1}{2} y^0_{\gamma,c,\Omega} - (\overline{y} \circ \Phi_t^\theta), v \det(\partial\Phi_t^\theta) \right)_{L^2(D)}. \end{aligned} \tag{3.27}$$

Note that $\det(\partial\Phi_t^\theta)$ can not be 0. Since $c_t \geq 0$ the bilinear form $\widetilde{a}_t(q, v) := a_t(q, v) + c(c_t q, v)_{L^2(D)}$ is continuous and coercive as a_t is also continuous and coercive and we can apply the Lemma of Lax-Milgram. Thus we have a unique solution q^t for every $t \in [0, \tau]$ to our problem (3.27) which means that $Y(t, y^t_{\gamma,c,\Omega_t}, y^0_{\gamma,c,\Omega})$ is single-valued for every $t \in [0, \tau]$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Ad (H3). We start by noting that the operator $\widetilde{\boldsymbol{M}}_\gamma$ defined by

$$\widetilde{\boldsymbol{M}}_\gamma : H^1(D) \to L^2(D) : y \mapsto \max_\gamma(\overline{\lambda} + c(y - \varphi)),$$

which is implicitly used in the proof of Theorem 3.6 is monotone due to Assumption 3.2 *(iii)*. So by definition we have for all $y_1, y_2 \in L^2(D)$

$$\Big(y_1 - y_2, \widetilde{\boldsymbol{M}}_\gamma(y_1) - \widetilde{\boldsymbol{M}}_\gamma(y_2)\Big)_{L^2(D)} \geq 0.$$

Specifically setting $y_2 \equiv 0$ we obtain

$$\big(\max_\gamma(\overline{\lambda} + c(y_1 - \varphi)), y_1\big)_{L^2(D)} \geq \big(\max_\gamma(\overline{\lambda} - c\varphi), y_1\big)_{L^2(D)} \qquad \forall y_1 \in L^2(D).$$

In the following we show that $\|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)}$ and $\|q^t\|_{H^1_0(D)}$, where q^t solves (3.27), is bounded by some constant C independent of t for $t \in [0, \tau]$. Since $y^t_{\gamma,c,\Omega_t} \in H^1_0(D)$, using coercivity of $a_t(.,.)$ with constant μ_t and using y^t_{γ,c,Ω_t} also as a test function we have

$$\begin{aligned}
0 \leq \|y^t_{\gamma,c,\Omega_t}\|^2_{H^1_0(D)} &\leq \mu_t a_t(y^t_{\gamma,c,\Omega_t}, y^t_{\gamma,c,\Omega_t}) \\
&= \mu_t \int_D (f_{\Omega_t} \circ \Phi^\theta_t) y^t_{\gamma,c,\Omega_t} \\
&\qquad - \max_\gamma(\overline{\lambda} \circ \Phi^\theta_t + c y^t_{\gamma,c,\Omega_t} - \varphi \circ \Phi^\theta_t)) y^t_{\gamma,c,\Omega_t} \, \mathrm{d}x \\
&\leq \mu_t \|f_{\Omega_t} \circ \Phi^\theta_t\|_{L^2(D)} \|y^t_{\gamma,c,\Omega_t}\|_{L^2(D)} \\
&\quad + \mu_t \|\max_\gamma(\overline{\lambda} \circ \Phi^\theta_t - c\varphi \circ \Phi^\theta_t)\|_{L^2(D)} \|y^t_{\gamma,c,\Omega_t}\|_{L^2(D)} \\
&\leq \mu_t \|f_{\Omega_t} \circ \Phi^\theta_t\|_{L^2(D)} \|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)} \\
&\quad + \mu_t \|\max_\gamma(\overline{\lambda} \circ \Phi^\theta_t - c\varphi \circ \Phi^\theta_t)\|_{L^2(D)} \|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)}.
\end{aligned}$$

Dividing both sites by $\|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)}$ we can, since Φ^θ_t converges to the identity as $t \to 0$ and $0 < \sup_{t\in[0,\tau]} \mu_t < +\infty$, further estimate

$$\begin{aligned}
\|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)} &\leq \sup_{t\in[0,\tau]} \Big(\mu_t \|f_{\Omega_t} \circ \Phi^\theta_t\|_{L^2(D)} + \mu_t \|\max_\gamma(\overline{\lambda} \circ \Phi^\theta_t - c\varphi \circ \Phi^\theta_t)\|_{L^2(D)}\Big) \\
&=: C < \infty. \qquad (3.28)
\end{aligned}$$

For q^t we calculate analogously using $\max'_\gamma \in [0, 1]$, $(\max'_\gamma(.)q^t, q^t)_{L^2(D)} \geq 0$, $c > 0$ and $\|y^t_{\gamma,c,\Omega_t}\|_{H^1_0(D)} \leq C$

$$\begin{aligned}
0 \leq \|q^t\|^2_{H^1_0(D)} &\leq \mu_t a_t(q^t, q^t) \\
&= -\mu_t \Big(c \int_0^1 \max'_\gamma(\overline{\lambda} \circ \Phi^\theta_t \\
&\qquad + c(s y^t_{\gamma,c,\Omega_t} + (1-s) y^0_{\gamma,c,\Omega} - \varphi \circ \Phi^\theta_t)) q^t \, \mathrm{d}s, q^t \det(\partial \Phi^\theta_t)\Big)_{L^2(D)} \\
&\quad - \mu_t \Big(\frac{1}{2} y^t_{\gamma,c,\Omega_t} + \frac{1}{2} y^0_{\gamma,c,\Omega} - (\overline{y} \circ \Phi^\theta_t), q^t \det(\partial \Phi^\theta_t)\Big)_{L^2(D)}
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$
\begin{aligned}
&\leq \mu_t \sup_{t\in[0,\tau]} (\det(\partial\Phi_t^\theta))\Big(\frac{1}{2}\|y^t_{\gamma,c,\Omega_t}\|_{L^2(D)} + \frac{1}{2}\|y^0_{\gamma,c,\Omega}\|_{L^2(D)} \\
&\qquad\qquad + \|(\overline{y}\circ\Phi_t^\theta)\|_{L^2(D)}\Big)\|q^t\|_{L^2(D)} \\
&\leq \mu_t \sup_{t\in[0,\tau]} (\det(\partial\Phi_t^\theta))\Big(\frac{1}{2}C + \frac{1}{2}\|y^0_{\gamma,c,\Omega}\|_{L^2(D)} \\
&\qquad\qquad + \sup_{t\in[0,\tau]} \|(\overline{y}\circ\Phi_t^\theta)\|_{L^2(D)}\Big)\|q^t\|_{H_0^1(D)}.
\end{aligned}
$$

Dividing both sides by $\|q^t\|_{H_0^1(D)}$ we obtain some upper bound $\|q^t\|_{H_0^1(D)} \leq C_q < \infty$ for $t \in [0,\tau]$. Now let any sequence $(t_n)_{n\in\mathbb{N}}$ with $t_n \to 0^+$ as $n \to +\infty$ be given. Since $(y^{t_n}_{\gamma,c,\Omega_{t_n}})_{n\in\mathbb{N}}$ is bounded in $H_0^1(D)$ we can find a subsequence $(t_{n_k})_{k\in\mathbb{N}}$ of $(t_n)_{n\in\mathbb{N}}$ such that $y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}} \rightharpoonup z$ in $H_0^1(D)$ for some $z \in H_0^1(D)$ as $k \to +\infty$. We now want to show that $z = y^0_{\gamma,c,\Omega}$.

To do so we want to apply Lebesgue's dominated convergence theorem on the state equation for $y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}}$ with $v \in H_0^1(D)$ as a test function

$$
a_t(y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}}, v) + \int_D \max_\gamma(\overline{\lambda}\circ\Phi_t^\theta + c(y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}} - \varphi\circ\Phi_t^\theta))v \,\mathrm{d}x - \int_D (f_{\Omega_t}\circ\Phi_t^\theta)v \,\mathrm{d}x = 0. \tag{3.29}
$$

Since the bilinear form $a_t(.,.)$ is bounded and $\|y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}}\|_{H_0^1(D)} \leq C$ we can apply Lebesgue's dominated convergence theorem to the underlying functions of $a_t(.,.)$.

Next we consider the $\max_\gamma$ term. Therefore let $z \in H_0^1(D)$. Then using boundedness of D, Assumption 3.2 *(ii)* and $\|z\|_{L^2(D)} \leq \|z\|_{H_0^1(D)}$ we have

$$
\begin{aligned}
\|\max_\gamma(\overline{\lambda} + c(z-\varphi))\|_{L^2(D)} &\leq \|\max_\gamma(\overline{\lambda} + c(z-\varphi)) \\
&\qquad - \max(0, \overline{\lambda} + c(z-\varphi))\|_{L^2(D)} \\
&\quad + \|\max(0, \overline{\lambda} + c(z-\varphi))\|_{L^2(D)} \\
&\leq \mathrm{vol}(D)^{1/2} g(\gamma) + \|\max(0, \overline{\lambda} + c(z-\varphi))\|_{L^2(D)} \\
&\leq \mathrm{vol}(D)^{1/2} g(\gamma) + \|\overline{\lambda} + c(z-\varphi)\|_{L^2(D)} \\
&\leq \mathrm{vol}(D)^{1/2} g(\gamma) + \|\overline{\lambda}\|_{L^2(D)} + c(\|z\|_{H_0^1(D)} + \|\varphi\|_{H_0^1(D)}) \\
&< +\infty.
\end{aligned}
$$

Using this inequality and (3.28) we get

$$
\begin{aligned}
&\|\max_\gamma(\overline{\lambda}\circ\Phi_t^\theta + c(y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}} - \varphi\circ\Phi_t^\theta))\|_{L^2(D)} \\
&\qquad \leq \mathrm{vol}(D)^{1/2} g(\gamma) + \sup_{t\in(t_{n_k})_{k\in\mathbb{N}}} \|\overline{\lambda}\circ\Phi_t^\theta\|_{L^2(D)} \\
&\qquad\quad + cC + c \sup_{t\in(t_{n_k})_{k\in\mathbb{N}}} \|\varphi\circ\Phi_t^\theta\|_{L^2(D)}.
\end{aligned}
$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Thus we can again apply Lebesgue's dominated convergence theorem. Since $y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}}$ is the unique solution of (3.29) for every $v \in H^1_0(D)$ and every $t_{n_k} \in [0,\tau]$ due to Theorem 3.6 we have $y^{t_{n_k}}_{\gamma,c,\Omega_{t_{n_k}}} \rightharpoonup y_{\gamma,c,\Omega}$ where $y_{\gamma,c,\Omega}$ is the unique solution of (3.11).

We now want to show the same kind of convergence for q^t, where q^t solves (3.27) for every $v \in H^1_0(D)$ in a similar way. Since we have already established an upper bound C_q for $\|q^t\|_{H^1_0(D)}$ we can again find a subsequence $(t_m)_{m\in\mathbb{N}}$ of $(t_{n_k})_{k\in\mathbb{N}}$ and some $u \in H^1_0(D)$ such that $q^{t_m} \rightharpoonup u$ in $H^1_0(D)$ as $m \to +\infty$.

Since the $\max'_\gamma(...)$ -term is continuous and bounded, the bilinear form $a_t(.,.)$ and $y^{t_m}_{\gamma,c,\Omega_{t_m}}$ are bounded and $y^{t_m}_{\gamma,c,\Omega_{t_m}}$ converges weakly we can once again apply Lebesgue's dominated convergence theorem, now on (3.27). The limit equation is our fully regularized adjoint equation (3.12), since $\max'_\gamma$ is continuous by Assumption 3.2 *(i)* and (3.12) has a unique solution $p_{\gamma,c,\Omega}$ due to Theorem 3.8. Thus we have $q^{t_m} \rightharpoonup p_{\gamma,c,\Omega}$ in $H^1_0(D)$.

From the proof of Theorem 3.14 we obtain that $\frac{\partial}{\partial t}G(t, y_{\gamma,c,\Omega}, q^{t_m})$ is weakly continuous in both its first and last argument. Thus we have

$$\lim_{\substack{m\to+\infty \\ t\to 0^+}} \partial_t G(t, y^0, q^{t_m}) = \partial_t G(0, y^0, p^0),$$

which is *(H3)*.

Thus we have shown that our problem satisifes all assumptions of Theorem 3.11. This gives us existence of the shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ given by

$$\mathrm{d}J_{\gamma,c}(\Omega)(\theta) = \frac{\mathrm{d}}{\mathrm{d}t}G(0, y_{\gamma,c,\Omega}, p_{\gamma,c,\Omega})$$

for $\theta \in C^1_c(D, \mathbb{R}^d)$. □

The following theorem gives the explicit form of the shape derivative $\mathrm{d}J_{\gamma,c}$ of our problem. It has been calculated before by [17, Theorem 3.4] but in a completely different way and with less general assumptions. Whereas the authors in [17] do not really take advantage of Theorem 3.11 for their formula we make full use of (3.22). Note also that our formula differs slightly from the one found in [17]. This is due to the fact that even though they look at the same problem as we do, they drop the assumption that f_Ω depends on the shape in their proofs. They pick up the assumption again later in [17, Remark 11].

Theorem 3.14. *Let the assumptions of Theorem 3.13 hold. Then the shape deriva-*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

tive $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ *is given by*

$$\begin{aligned}
\mathrm{d}J_{\gamma,c}(\Omega)(\theta) = \int_D & -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y}^T \theta) + \nabla y_{\gamma,c,\Omega}^T \nabla M \cdot \theta \nabla p_{\gamma,c,\Omega} \\
& - \nabla y_{\gamma,c,\Omega}^T \nabla \theta^T M \nabla p_{\gamma,c,\Omega} - \nabla y_{\gamma,c,\Omega}^T \nabla \theta M \nabla p_{\gamma,c,\Omega} \\
& + p_{\gamma,c,\Omega}((\nabla e^T \theta)^T \nabla y_{\gamma,c,\Omega} - e^T \nabla \theta \nabla y_{\gamma,c,\Omega}) \\
& + y_{\gamma,c,\Omega}((\nabla e^T \theta)^T \nabla p_{\gamma,c,\Omega} - e^T \nabla \theta \nabla p_{\gamma,c,\Omega}) + \nabla b \cdot \theta y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& + \max_\gamma' \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) (\nabla \overline{\lambda}^T \theta - c \nabla \varphi^T \theta) p_{\gamma,c,\Omega} - \dot{f}_\theta p_{\gamma,c,\Omega} \, \mathrm{d}x \\
+ \int_D & \operatorname{div}(\theta) \Bigg(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T M \nabla p_{\gamma,c,\Omega} \\
& \qquad + e^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& \qquad + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} - f_\Omega p_{\gamma,c,\Omega} \Bigg) \mathrm{d}x,
\end{aligned} \tag{3.30}$$

where $\nabla M \cdot \theta$ *denotes the matrix* $(\nabla m_{i,j}^T \theta)_{i,j}$.

Proof. We have to calculate $\partial_t G(0, y_{\gamma,c,\Omega}^0, p_{\gamma,c,\Omega}^0)$. Using integration by substitution we can write (3.23) as

$$\begin{aligned}
G(t,y,p) = & \frac{1}{2} \int_D \left(y - \overline{y} \circ \Phi_t^\theta\right)^2 \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x \\
& + \int_D \nabla y^T \partial(\Phi_t^\theta)^{-T} (M \circ \Phi_t^\theta) \partial(\Phi_t^\theta)^{-1} \nabla p \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x \\
& + \int_D (e \circ \Phi_t^\theta)^T \left(p \partial(\Phi_t^\theta)^{-1} \nabla y + y \partial(\Phi_t^\theta)^{-1} \nabla p\right) \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x \\
& + \int_D (b \circ \Phi_t^\theta) y p \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x \\
& + \int_D \max_\gamma \left(\overline{\lambda} \circ \Phi_t^\theta + c(y - \varphi \circ \Phi_t^\theta)\right) p \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x \\
& - \int_D (f_{\Omega_t} \circ \Phi_t^\theta) p \det\left(\partial \Phi_t^\theta\right) \, \mathrm{d}x.
\end{aligned}$$

We now differentiate each term individually with respect to t. The following swaps of the integration and differentiation are justified since all integrands are integrable, differentiable and the derivatives are bounded. Using the chainrule, Definition 3.12

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and the theorems from Section 2.3 we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}G(t,y,p)\big|_{t=0} = &\int_D -(y-\overline{y})(\nabla\overline{y}\cdot\theta) + \frac{1}{2}(y-\overline{y})^2 \operatorname{div}(\theta)\ \mathrm{d}x \\
&+ \int_D \nabla y M \nabla p \operatorname{div}(\theta) + \nabla y^T \nabla M \cdot \theta \nabla p - \nabla y^T \nabla \theta^T M \nabla p \\
&\qquad - \nabla y^T \nabla \theta M \nabla p\ \mathrm{d}x \\
&+ \int_D e^T(p\nabla y + y\nabla p) \operatorname{div}(\theta) + (\nabla e^T \theta)^T (\nabla y p + y_{\gamma,c,\Omega}\nabla p) \\
&\qquad - e^T(p\nabla\theta\nabla y + y\nabla\theta\nabla p)\ \mathrm{d}x \\
&+ \int_D byp \operatorname{div}(\theta) + \nabla b \cdot \theta y p\ \mathrm{d}x \\
&+ \int_D \max_\gamma'\left(\overline{\lambda} + c(y-\varphi)\right)(\nabla\overline{\lambda}\cdot\theta - c\nabla\varphi\cdot\theta)p \\
&\qquad + \max_\gamma\left(\overline{\lambda} + c(y-\varphi)\right) p \operatorname{div}(\theta)\ \mathrm{d}x \\
&- \int_D \dot{f}_\theta p + f_\Omega p \operatorname{div}(\theta)\ \mathrm{d}x.
\end{aligned}$$

By rearranging the individual terms and evaluating the right-hand side at $y = y_{\gamma,c,\Omega}$ and $p = p_{\gamma,c,\Omega}$ we obtain the claimed result. □

Remark 3.15. Note that our shape derivative formula differs from the one found in [17, Theorem 3.4]. We have $\int_D \max_\gamma'(\lambda + c(y_{\gamma,c,\Omega} - \varphi))(\nabla\lambda^T\theta)p_{\gamma,c,\Omega}\ \mathrm{d}x$ as an additional term in our shape derivative which can not be found in [17]. The authors of that paper claim that the material derivative $\dot{\overline{\lambda}}$ of $\overline{\lambda}$ is 0. We do not see why that would be the case and therefore do not neglect this additional term.

For PDE constrained shape derivatives as introduced in Example 2.8 there are usually two equivalent shape derivative formulas. The first one considers the whole shape Ω and would look similar to (3.30). The second one only depends on the normal component of the vector field on the boundary $\partial\Omega$ of the shape Ω. This is because all the relevant information for the shape derivative is encoded in how the vector field deforms the boundary. For more details see e.g. [5, Chapter 10, Section 5.6]

In our setting such a shape derivative which only depends on the boundary can not exist in general. This is due to the material derivative $\dot{f}_\theta$, which shows up in (3.30). When the material derivative vanishes though, as is the case for a piece-wise constant f_Ω for example, we can indeed find a boundary expression for the shape derivative.

Theorem 3.16. *Let f_Ω be piece-wise constant and let $\theta \in C_c^1(D,\mathbb{R}^d)$. Then the material $\dot{f}_\theta$ satisifies $\dot{f}_\theta \equiv 0$ a.e. in D.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. We split the integrals over D in integrals over Ω and Ω^c and apply the definition of the material derivaitve on the two subdomains separately. For $x \in \Omega$ we have $\Phi_t^\theta(x) \in \Omega_t$ for $\theta \in C_c^1(\overline{D}, \mathbb{R}^d)$ and thus by definition of f_Ω

$$\dot{f}_\theta = \lim_{t\to 0+} \frac{(f_{\Omega_t} \circ \Phi_t^\theta)(x) - f_\Omega(x)}{t} = \lim_{t\to 0+} \frac{f_{\Omega_t}(\Phi_t^\theta(x)) - f_\Omega(x)}{t} = \lim_{t\to 0+} \frac{0}{t} = 0$$

and the same holds true for $x \in \Omega^c$. □

Before we can state the following theorem we need a short introduction to the dyadic product $a \otimes b$ between two vectors $a, b \in \mathbb{R}^d$. It is defined by

$$(a \otimes b)c := (b \cdot c)a \quad \text{for all } c \in \mathbb{R}^d.$$

Let $A, B \in \mathbb{R}^{d,d}$ and $A : B = \sum_{i=1}^d \sum_{j=1}^d a_{i,j} b_{i,j}$ be the Frobenius inner product. Then it holds for $A \in \mathbb{R}^{d,d}$ and $b, c \in \mathbb{R}^d$ that $A : (b \otimes c) = b \cdot Ac$. For further details see [22, Section 6.3].

Theorem 3.17. *Let $D \subset \mathbb{R}^d$ be an open, bounded domain with Lipschitz boundary ∂D, the bilinear form $a(.,.)$ be as in Section 3.1 with $M = Id$ and $(e_i)_i$, b in $L^\infty(D) \cap H^1(D)$ and let f_Ω be piece-wise constant. Let $c > 0$, $\varphi \in H^1(D)$ and $\overline{\lambda} \in H^1(D)$. Let y_Ω be the solution to (3.4) with $\partial\Omega \in C^2$. Let $\theta \in C_c^2(D, \mathbb{R}^d)$.*

Then the material derivative $dJ_{\gamma,c}(\Omega)(\theta)$ is given by

$$\mathrm{d}J_{\gamma,c}(\Omega)(\theta) = \int_{\partial\Omega} S_1 \nu \cdot \theta \mathrm{d}s,$$

where ν is the unit normal vector along $\partial\Omega$ and S_1 is given by

$$\begin{aligned} S_1 := &- \nabla p_{\gamma,c,\Omega} \otimes \nabla y_{\gamma,c,\Omega} - \nabla y_{\gamma,c,\Omega} \otimes \nabla p_{\gamma,c,\Omega} - p_{\gamma,c,\Omega} e \otimes \nabla y_{\gamma,c,\Omega} - y_{\gamma,c,\Omega} e \otimes \nabla p_{\gamma,c,\Omega} \\ &+ \Big(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T \nabla p_{\gamma,c,\Omega} + e^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) \\ &\quad + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} - f_\Omega p_{\gamma,c,\Omega} \Big) Id \end{aligned}$$

Proof. First we note that under our assumptions the shape derivative (3.30) simplifies

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to

$$
\begin{aligned}
dJ_{\gamma,c}(\Omega)(\theta) = \int_D & -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y}^T \theta) - \nabla y_{\gamma,c,\Omega}^T \nabla \theta^T \nabla p_{\gamma,c,\Omega} \\
& - \nabla y_{\gamma,c,\Omega}^T \nabla \theta \nabla p_{\gamma,c,\Omega} \\
& + p_{\gamma,c,\Omega}((\nabla e^T \theta)^T \nabla y_{\gamma,c,\Omega} - e^T \nabla \theta \nabla y_{\gamma,c,\Omega}) \\
& + y_{\gamma,c,\Omega}((\nabla e^T \theta)^T \nabla p_{\gamma,c,\Omega} - e^T \nabla \theta \nabla p_{\gamma,c,\Omega}) + \nabla b \cdot \theta y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& + \max_\gamma' \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) (\nabla \overline{\lambda}^T \theta - c \nabla \varphi^T \theta) p_{\gamma,c,\Omega} \; dx \\
+ \int_D & \operatorname{div}(\theta) \Bigg(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T \nabla p_{\gamma,c,\Omega} \\
& \qquad + e^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& \qquad + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} - f_\Omega p_{\gamma,c,\Omega} \Bigg) dx.
\end{aligned}
$$

Using $\operatorname{div}(\theta) = \mathrm{Id} : \nabla \theta$ and the previously introduced dyadic notations we obtain

$$
\begin{aligned}
\mathrm{d}J_{\gamma,c}(\Omega)(\theta) = \int_D & -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y} \cdot \theta) - \nabla p_{\gamma,c,\Omega} \otimes \nabla y_{\gamma,c,\Omega} : \nabla \theta \\
& - \nabla y_{\gamma,c,\Omega} \otimes \nabla p_{\gamma,c,\Omega} : \nabla \theta \\
& + p_{\gamma,c,\Omega}((\nabla y_{\gamma,c,\Omega}^T \nabla e^T) \cdot \theta - e \otimes \nabla y_{\gamma,c,\Omega} : \nabla \theta) \\
& + y_{\gamma,c,\Omega}((\nabla p_{\gamma,c,\Omega}^T \nabla e^T) \cdot \theta - e \otimes \nabla p_{\gamma,c,\Omega} : \nabla \theta) + y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \nabla b \cdot \theta \\
& + \max_\gamma' \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} (\nabla \overline{\lambda}^T - c \nabla \varphi^T) \cdot \theta \; dx \\
+ \int_D & \Bigg(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T \nabla p_{\gamma,c,\Omega} \\
& \qquad + e^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& \qquad + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} - f_\Omega p_{\gamma,c,\Omega} \Bigg) \mathrm{Id} : \nabla \theta \, dx.
\end{aligned}
$$

Rearranging the terms we can write this expression as

$$
\mathrm{d}J_{\gamma,c}(\Omega)(\theta) = \int_D S_1 : \nabla \theta + S_0 \cdot \theta \; \mathrm{d}x,
$$

where S_1 and S_2 are given by

$$
\begin{aligned}
S_1 := & - \nabla p_{\gamma,c,\Omega} \otimes \nabla y_{\gamma,c,\Omega} - \nabla y_{\gamma,c,\Omega} \otimes \nabla p_{\gamma,c,\Omega} - p_{\gamma,c,\Omega} e \otimes \nabla y_{\gamma,c,\Omega} - y_{\gamma,c,\Omega} e \otimes \nabla p_{\gamma,c,\Omega} \\
& + \Bigg(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T \nabla p_{\gamma,c,\Omega} + e^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) \\
& \qquad + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} - f_\Omega p_{\gamma,c,\Omega} \Bigg) \mathrm{Id}
\end{aligned}
$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and

$$S_0 := -\left(y_{\gamma,c,\Omega} - \overline{y}\right)\nabla\overline{y} + p_{\gamma,c,\Omega}\nabla y^T_{\gamma,c,\Omega}\nabla e^T + y_{\gamma,c,\Omega}\nabla p^T_{\gamma,c,\Omega}\nabla e^T + y_{\gamma,c,\Omega}p_{\gamma,c,\Omega}\nabla b$$
$$+ \max'_\gamma\left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega}(\nabla\overline{\lambda}^T - c\nabla\varphi^T).$$

Due to our assumed additional regularity of our functions and boundary we have $y_{\gamma,c,\Omega},\ p_{\gamma,c,\Omega} \in H^2(D) \cap H^1_0(D)$, see [7]. Therefore $S_1 \in W^{1,1}(D)$. Since the shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ is 0 if the vector field θ does not change the boundary we have

$$\int_D S_1 : \nabla\theta + S_0 \cdot \theta \ \mathrm{d}x = 0 \quad \text{for all } \theta \in C^2(D, \mathbb{R}^d) \text{ with compact support in } D\backslash\partial\Omega.$$

Partial integration yields

$$\int_D (-\operatorname{div}(S_1) + S_0) \cdot \theta \ \mathrm{d}x = 0 \quad \text{for all } \theta \in C^2(D, \mathbb{R}^d) \text{ with compact support in } D\backslash\partial\Omega$$

and by applying the fundamental theorem of calculus of variations we obtain that $-\operatorname{div}(S_1) + S_0 = 0$ a.e. in $D\backslash\partial D$.

Finally by partial integration

$$\begin{aligned}
\mathrm{d}J_{\gamma,c}(\Omega)(\theta) &= \int_D S_1 : \nabla\theta + S_0 \cdot \theta \ \mathrm{d}x \\
&= \int_D -\operatorname{div}(S_1) + S_0) \cdot \theta \ \mathrm{d}x + \int_{\partial\Omega} S_1\nu \cdot \theta \ \mathrm{d}s \\
&= \int_{\partial\Omega} S_1\nu \cdot \theta \ \mathrm{d}s
\end{aligned}$$

we obtain the result. □

So far in our problem only f_Ω has been explicitly dependent on the shape Ω. A very natural question which arises now is, whether we can extend our results further and make our obstacle φ or our bilinear form $a(.,.)$ explicitly depend on our shape. To this the answer is yes if we assume $a_\Omega(.,.)$ to fulfill all assumptions above for every Ω in our set of admissible functions. The case of a shape dependent obstacle φ_Ω has already been discussed in [17, Remark 11]. If we carefully reconsider the arguments made we notice, that shape-dependence of φ, M, e or b makes no real difference in the arguments given. Only the shape derivative itself changes, as we have material derivatives $\dot{.}_\theta$ instead of $\nabla . \cdot \theta$. So the shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)$ for shape-dependent

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

M_Ω, e_Ω, b_Ω, φ_Ω and f_Ω is given by

$$
\begin{aligned}
dJ_{\gamma,c}(\Omega)(\theta) = \int_D & -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y}^T \theta) + \nabla y_{\gamma,c,\Omega}^T \dot{M}_\theta \nabla p_{\gamma,c,\Omega} \\
& - \nabla y_{\gamma,c,\Omega}^T \nabla \theta^T M_\Omega \nabla p_{\gamma,c,\Omega} - \nabla y_{\gamma,c,\Omega}^T \nabla \theta M_\Omega \nabla p_{\gamma,c,\Omega} \\
& + p_{\gamma,c,\Omega}(\dot{e}_\theta^T \nabla y_{\gamma,c,\Omega} - e_\Omega^T \nabla \theta \nabla y_{\gamma,c,\Omega}) \\
& + y_{\gamma,c,\Omega}(\dot{e}_\theta^T \nabla p_{\gamma,c,\Omega} - e_\Omega^T \nabla \theta \nabla p_{\gamma,c,\Omega}) \\
& + \dot{b}_\theta y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& + \max_\gamma' \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) (\nabla \overline{\lambda}^T \theta - c\dot{\varphi}_\theta) p_{\gamma,c,\Omega} \\
& - \dot{f}_\theta p_{\gamma,c,\Omega} \; dx \\
+ \int_D & \operatorname{div}(\theta) \left(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T M_\Omega \nabla p_{\gamma,c,\Omega} \right. \\
& \quad + e_\Omega^T (p_{\gamma,c,\Omega} \nabla y_{\gamma,c,\Omega} + y_{\gamma,c,\Omega} \nabla p_{\gamma,c,\Omega}) \\
& \quad + b y_{\gamma,c,\Omega} p_{\gamma,c,\Omega} \\
& \quad + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} \\
& \quad \left. - f_\Omega p_{\gamma,c,\Omega} \right) dx,
\end{aligned}
$$

where $\dot{M}_\theta$ denotes the matrix $(\dot{m}_{i,j,\theta})_{i,j}$ and $\dot{e}_\theta$ the vector $(\dot{e}_{i,\theta})_i$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4. Numerical Experiments for the Fully Regularized Problem

In this chapter we demonstrate the theoretical results from Chapter 3 with a concrete example. We present numerical experiments using Netgen/NGSolve, a finite element software which comes with both a mesh generation tool and a solver for variational formulations. A lot of the development for the software is done at TU Wien and in recent versions a fully automated shape differentiation functionality was added, see [10].

NGSolve uses a Python interface while the code itself is written in C++. Unlike in [17], where the focus of the experiments lies on demonstrating the previously shown results, we are more interested in comparing the shape derivative formula with NGSolve's internal shape differentiation functionality.

The outline of this chapter is as follows: In Section 4.1 we introduce our experiment setup, which is the same as in [17]. We perform some further analysis to show why the used example is valid in view of the assumptions made in Chapter 3 even though our function f_Ω has less regularity. Next we numerically calculate our target function $\overline{y}$ where we investigate the choice of $\overline{\lambda}$ introduced in (3.10). In Section 4.2.2 we take a look at how well (3.30) performs compared to NGSolve's shape differentiation functionality. Finally we perform a shape optimization experiment where we mostly investigate the influence of the chosen bilinear form to get a suitable vector field and again compare formula (3.30) with NGSolve's internal functions.

4.1. Experiment setup

In this section we formulate the problem for our numerical experiments. We look at the same problem as in [17] so we can compare our results. Their experiments were done in FEniCS, a finite element package for Python whereas our calculations are carried out in NGSolve. Since NGSolve comes with its own shape derivative functionality we are particularly interested in the differences between our shape derivative formula (3.30) and its counterpart in NGSolve. Thus also the focus of the experiments is slightly different to [17] as we take a closer look in NGSolve specific questions.

Given $D = [0,1]^2 \subset \mathbb{R}^2$, an obstacle $\varphi \in L^1_{loc}(D)$, a target function $\overline{y} \in H^1(D)$ and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the function $f_\Omega \in L^2(D)$ defined by

$$f_\Omega(x) := \begin{cases} -10 & \text{for } x \in D\backslash\Omega, \\ 100 & \text{for } x \in \Omega \end{cases} \tag{4.1}$$

we want to solve the following optimization problem

$$\min_\Omega \frac{1}{2} \int_D |y_\Omega - \overline{y}|^2 \, \mathrm{d}x, \tag{4.2}$$

subject to $y_\Omega \in H_0^1(D)$ satisfies for some $\lambda \in L^2(D)$

$$\begin{aligned} \int_D \nabla y_\Omega^T \nabla v \, \mathrm{d}x + (\lambda, v)_{L^2(D)} &= \int_D f_\Omega v \, \mathrm{d}x \quad \text{for all } v \in H_0^1(D) \\ \lambda &\geq 0 \quad \text{in } D, \\ y_\Omega &\leq \varphi \quad \text{in } D, \\ \lambda(y_\Omega - \varphi) &= 0 \quad \text{in } D. \end{aligned} \tag{4.3}$$

To formulate the fully regularized state and adjoint equation we need a smoothed $\max(0, .)$ with a continuous derivative. As in [17] we use for $\gamma > 0$

$$\max_\gamma(x) := \begin{cases} \max(0, x) & \text{for } x \in \mathbb{R}\backslash[-\frac{1}{\gamma}, \frac{1}{\gamma}], \\ \frac{\gamma}{4}x^2 + \frac{1}{2}x + \frac{1}{4\gamma} & \text{for } x \in (-\frac{1}{\gamma}, \frac{1}{\gamma}) \end{cases}$$

and its derivative

$$\max'_\gamma(x) := \begin{cases} 0 & \text{for } x \in (-\infty, -\frac{1}{\gamma}], \\ \frac{\gamma}{2}x + \frac{1}{2} & \text{for } x \in (-\frac{1}{\gamma}, \frac{1}{\gamma}), \\ 1 & \text{for } x \in [\frac{1}{\gamma}, +\infty). \end{cases}$$

This particular choice of $\max_\gamma$ satisfies all assumptions proposed in Assumption 3.2. We test our results on the two obstacles

$$\varphi_1(x) := 0.5 \qquad \text{and} \qquad \varphi_2(x) := 5e^{-x_1 - 1}. \tag{4.4}$$

Note that in our setting we have $\dot{f}_\theta \equiv 0$ due to Theorem 3.16. In this setting the fully regularized state equation is given by

$$\int_D \nabla y_{\gamma,c,\Omega}^T \nabla v + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) v - f_\Omega v \, \mathrm{d}x = 0 \quad \forall v \in H_0^1(D), \tag{4.5}$$

and the fully regularized adjoint equation by

$$\int_D \nabla p_{\gamma,c,\Omega}^T \nabla v + c \max'_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} v + (y_{\gamma,c,\Omega} - \overline{y}) v \, \mathrm{d}x \quad \forall v \in H_0^1(D).$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ finally simplifies to

$$\begin{aligned} dJ_{\gamma,c}(\Omega)(\theta) = \int_D & -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y}^T \theta) \\ & - \nabla y_{\gamma,c,\Omega}^T (\nabla \theta^T + \nabla \theta) \nabla p_{\gamma,c,\Omega} \\ & + \max_\gamma' \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) (\nabla \overline{\lambda}^T \theta - c \nabla \varphi^T \theta) p_{\gamma,c,\Omega} \, \mathrm{d}x \\ + \int_D & \operatorname{div}(\theta) \left(\frac{1}{2}(y_{\gamma,c,\Omega} - \overline{y})^2 + \nabla y_{\gamma,c,\Omega}^T \nabla p_{\gamma,c,\Omega} \right. \\ & \qquad + \max_\gamma \left(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi)\right) p_{\gamma,c,\Omega} \\ & \qquad \left. - f_\Omega p_{\gamma,c,\Omega} \right) \mathrm{d}x. \end{aligned} \tag{4.6}$$

4.2. Experiments

In this section we will take a look at the experiments performed. The respective Python code can be found in Appendix A.

4.2.1. The Target Function and Choice of $\overline{\lambda}$

To solve our shape optimization problem (4.2) we need a target function which we want to approximate. While we could of course choose any function $\overline{y} \in H_0^1(D)$ and approximate it by a function $y_{\gamma,c,\Omega}$ satisfying (4.3) we also want to see how good our approximation actually is. Thus we choose a $\overline{y} \in H_0^1(D)$ which itself satisfies (4.3). This leaves us with defining a target shape Ω_{target}, which we choose the same as in [17]. It is defined by $\Omega_{\text{target}} := (C_1 \cup C_2 \cup C_3) \backslash C_4$ where the sets $C_1,\ C_2,\ C_3,\ C_4$ are given by

$$\begin{aligned} C_1 &:= \left\{(x_1, x_2)^T \in [0,1]^2 : (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.3 \text{ and } x_1 \geq 0.5\right\}, \\ C_2 &:= \left\{(x_1, x_2)^T \in [0,1]^2 : (x_1 - 0.5)^2 + (x_2 - 0.7)^2 \leq 0.1\right\}, \\ C_3 &:= \left\{(x_1, x_2)^T \in [0,1]^2 : (x_1 - 0.5)^2 + (x_2 - 0.3)^2 \leq 0.1\right\}, \\ C_4 &:= \left\{(x_1, x_2)^T \in [0,1]^2 : (x_1 - 0.5)^2 + (x_2 - 0.5)^2 < 0.1\right\}. \end{aligned}$$

It is also depicted in Figure 4.1.

In our first experiment we want to investigate the influence of the $\overline{\lambda}$ we choose in order to solve (3.11) and (3.12). Solving the fully regularized state equation requires solving a non-linear problem. Thus we use an iterative Newton solver. Since the target function $\overline{y}_{\gamma,c}$ should also satisfy the fully regularized state equation it must satisfy $\overline{y}_{\gamma,c} \leq \varphi$. While every choice of $\overline{\lambda} \in L^2(D)$ guarantees convergence to the unregularized solution $\overline{y}$ as $\gamma, c \to +\infty$ not every $\overline{\lambda}$ will guarantee convergence from the right direction, i.e. $\overline{y}_{\gamma,c_1} \leq \overline{y}_{\gamma,c_2} \leq \overline{y} \leq \varphi$ for all $0 < c_1 \leq c_2 \leq +\infty$. In

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

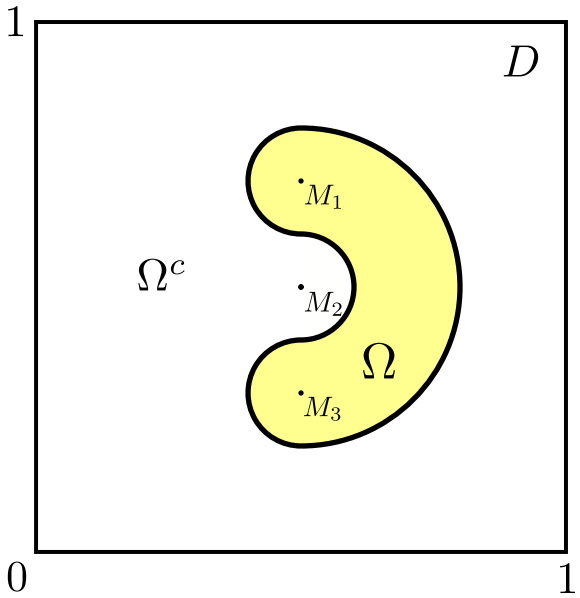


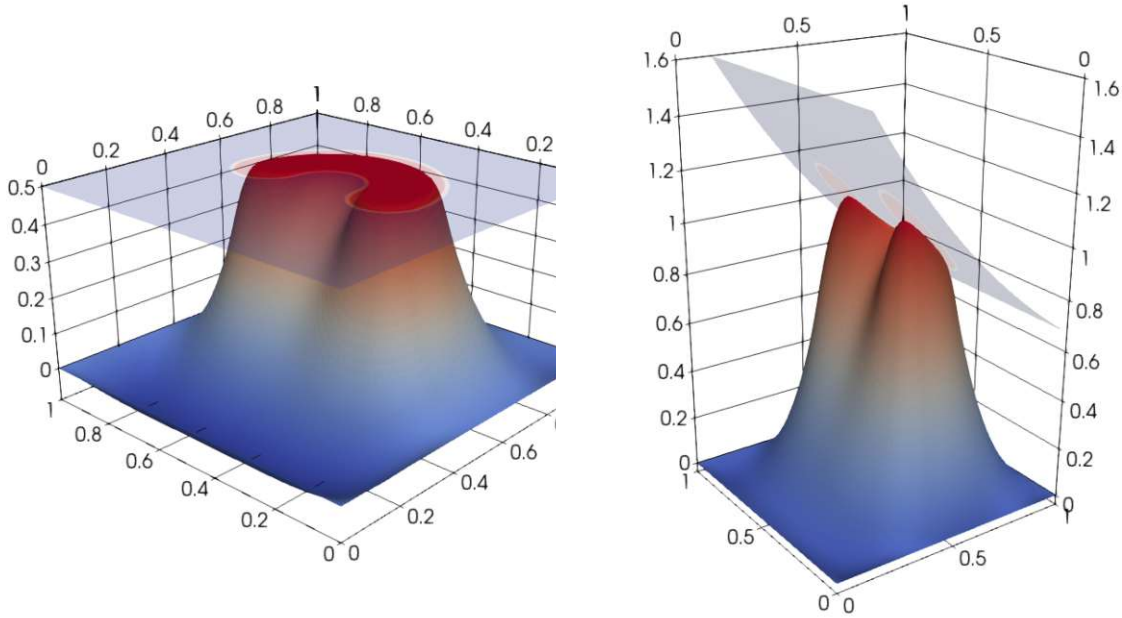
Figure 4.1.: The hold-all domain D with Ω^c and the shape Ω used to calculate $\overline{y}$. It consists of the half-circle of radius 0.3 around $M_2 = (0.5, 0.5)$ plus the two half-circles of radius 0.1 around $M_1 = (0.5, 0.7)$ and $M_3 = (0.5, 0.3)$ minus the half-circle of radius 0.1 around M_2.

[13, (3.10)] the authors propose the choice of $\overline{\lambda} = \max(0, f - \mathbf{S}\varphi)$ where $\mathbf{S}$ is the differential operator corresponding to our bilinear form $a(.,.)$. In our setting $\mathbf{S}$ is the Laplace operator Δ. We will compare this choice with $\overline{\lambda} \equiv 0$, $\overline{\lambda} = 100(x_1(1 - x_1))^2(x_2(1-x_2))^2 \in H_0^1(D)$ and $\overline{\lambda} = \max(0, -\mathbf{S}\varphi)$. The code for this experiment can be found in Appendix A.2.

To generate the target function in NGSolve we first generate the boundary ∂D and the interface $\Gamma = \partial\Omega$ using splines. This way we can represent the boundary exactly. We generate a triangular mesh with a maximum element diameter of 0.01. We define $\gamma = 500$ and loop through $c = 2^i$, $i = 0, ..., 17$. We use a finite element space (FES) of order 1 and define our non-linear variational formulation. To solve this non-linear problem we use the multidimensional Newton's method.

In Figure 4.3 we see the convergence of $\overline{y}_{\gamma,c}((0.7, 0.5)^T)$ which touches the obstacle in the limit $\gamma, c \to \infty$ for various $\overline{\lambda}$. We have convergence for all $\overline{\lambda}$ but only for $\overline{\lambda} = \max(0, f_\Omega - \Delta\varphi)$ we have $\overline{y}_{\gamma,c} \leq \varphi$ for all $c > 0$. The convergence plots for the other $\overline{\lambda}$ are almost indistinguishable. Increasing c comes at a cost. For larger c the number of iterations needed for the Newton's method to converge increases. In Figure 4.2 we see the target functions for both our obstacles.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



(a) $\overline{y}$ with obstacle φ_1

(b) $\overline{y}$ with obstacle φ_1

Figure 4.2.: The target functions $\overline{y}$ with obstacles φ_1 and φ_2. The color scheme on the obstacle indicates the target shape. Note that obstacle φ_2 is clipped at $x_3 = 1.6$.

4.2.2. The Shape Derivative Formula

Next we want to test our shape derivative formula (4.6) and compare it to NGSolve's DiffShape() functionality. The code for this section can be found in Appendix A.3. In order to calculate (4.6) we need the solutions of both the fully regularized state and adjoint equation. To get these we have to solve two non-linear problems, each requiring the application of Newton's method. Once we have those we have all unknowns in (4.6) and can calculate the integral for any given vector field.

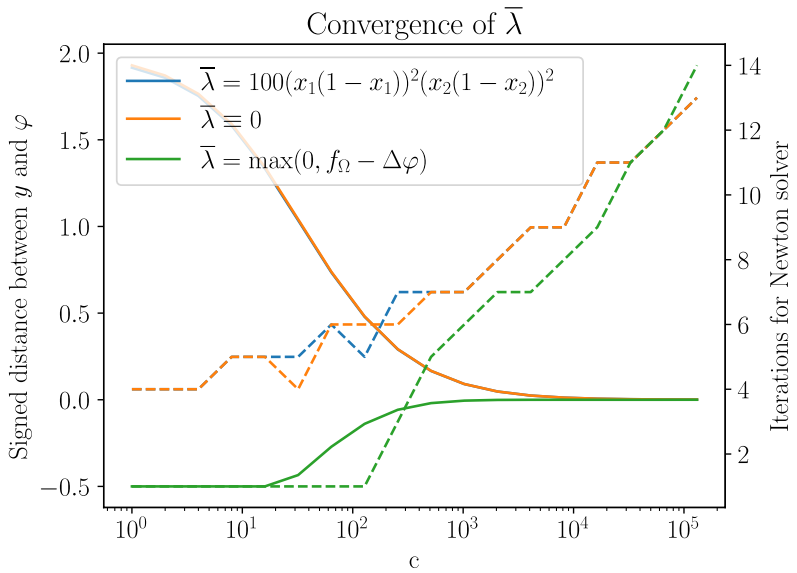
To use NGSolve's fully-automated DiffShape() we have to define the Lagrange function $\mathcal{L}$ in (3.20). Then to calculate the shape derivative we simply call the function DiffShape() on the Lagrangian $\mathcal{L}$. In our setting we get a wrong result though. As it turns out the DiffShape() function misses the term

$$\int_D -(y_{\gamma,c,\Omega} - \overline{y})(\nabla \overline{y}^T \theta) \, \mathrm{d}x \tag{4.7}$$

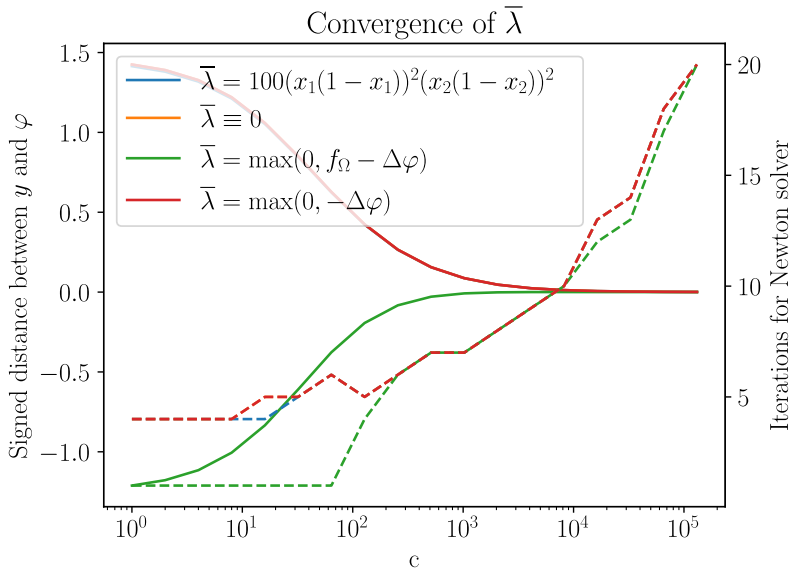
in (4.6). To explain this behavior we need to elaborate further on NGSolve.

To represent a function, e.g. $\varphi_2(x_1, x_2) = 5e^{-x_1-1}$, in NGSolve there are two

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



(a) Obstacle φ_1



(b) Obstacle φ_2

Figure 4.3.: The distance $\overline{y}_{\gamma,c} - \varphi$ evaluated at a point which touches the obstacle in the limit for various $c > 0$ and $\overline{\lambda}$. Only the right choice of $\overline{\lambda}$ guarantees $\overline{y}_{\gamma,c} \leq \varphi$ for all $c > 0$. The dashed lines indicate the number of iterations needed for the Newton solver to converge.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

different ways to do so. One is by means of a so called *coefficient function*. A coefficient function corresponds to the abstract expression $\varphi_2(x_1, x_2) := 5e^{-x_1-1}$, i.e. it is independent of any mesh. On the other hand there are *grid functions*. A grid function is a representation of a function in a FES. This means they have a certain order and live over a certain mesh. In NGSolve the user can convert any coefficient function into a grid function by projecting the function onto the FES. The other way round is not possible.

Our target function $\overline{y}_{\gamma,c}$ is the solution of a finite element problem and therefore we only got its representation in the FES, i.e. as a grid function. That is where the problem arises, since DiffShape() does not handle grid functions properly yet. Since all other functions (except $y_{\gamma,c,\Omega}$ and $p_{\gamma,c,\Omega}$, but they are arguments of the Lagrangian $\mathcal{L}$ and therefore handled differently) are coefficient functions it is sufficient in our case to simply add the term (4.7) to correct DiffShape().

To test the implementations of our shape derivative $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ we perform a Taylor test. We choose an arbitrary $\theta \in C_c^1(D, \mathbb{R}^d)$, in our case

$$\theta(x_1, x_2) := (10x_1^2(1-x_1)^2, 11x_2^2(1-x_2)^2)^T,$$

and calculate

$$\lim_{t\to 0+} |J(\Omega_t) - J(\Omega) - t\mathrm{d}J_{\gamma,c}(\Omega)(\theta)|,$$

where $\Omega_t = \Phi_t^\theta(\Omega)$.

As the maximum diameter for any element in our mesh we choose 0.03. The shape Ω is given by the circle of radius 0.3 around $M_2 = (0.5, 0.5)$. This time the FES is of order 3, $c = 1000$ and $\gamma = 50$. In Figure 4.4 we depict the results of our Taylor test with obstacle φ_2. We expect the error to converge to 0 with at least order 2 since we are looking at the local error. Both NGSolve's DiffShape() and our shape derivative formula (4.6) reach this order but NGSolve's DiffShape() seems to perform better. The integral formula stops converging earlier then NGSolve's DiffShape(). Further experiments showed that by decreasing the maximum element diameter in our mesh the point where this happens can be moved further down.

In Figure 4.5 we plot the difference between the corrected DiffShape() and the integral formula (4.6) for various maximum element diameters of our mesh. We set $c = 1000$, $\gamma = 50$ and φ_2 as the obstacle. Ω is a circle of radius 0.3 centered around $M_2 = (0.5, 0.5)$. In our experiment we see roughly a convergence of order 2 with respect to the maximum element diameter of our mesh.

4.2.3. Shape Convergence

We now want to numerically solve the actual shape optimization problem (4.2). To do so we want to use a gradient algorithm as found in [10, Algorithm 1]. In every step we move our domain Ω in a direction θ, which minimizes our shape functional J.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

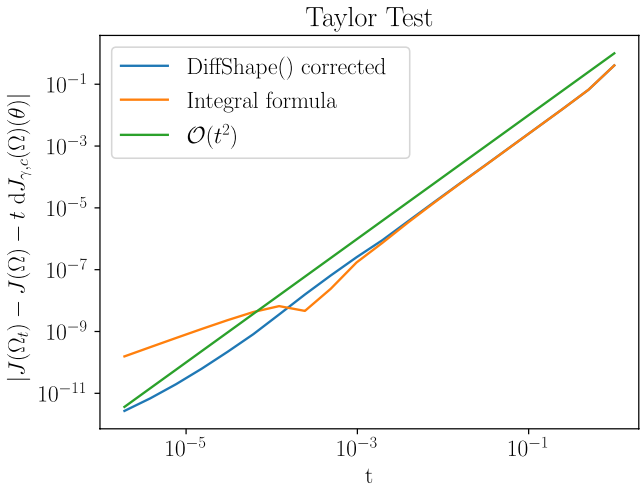


Figure 4.4.: Taylor test with obstacle φ_2, a maximum element diameter of our mesh of 0.03, $c = 1000$ and $\gamma = 50$. The shape Ω is given as the circle of radius 0.3 around $M_2 = (0.5, 0.5)$. For the corrected DiffShape() we added (4.7) to the result of DiffShape(). The integral formula (4.6) does not perform as good as the corrected DiffShape().

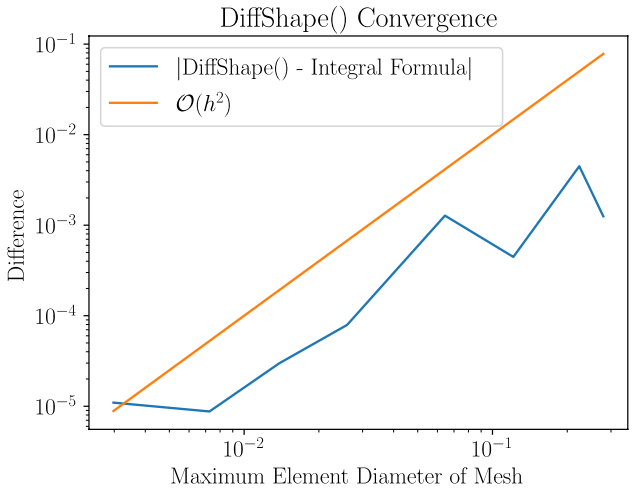


Figure 4.5.: Difference between NGSolve's corrected DiffShape() and the integral formula (4.6) for various maximum element diameters of our mesh. The obstacle is φ_2, $c = 1000$ and $\gamma = 50$. The shape Ω is given by the circle of radius 0.3 around $M_2 = (0.5, 0.5)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Thus we have to find a suitable θ in every step of our algorithm, which poses another problem to solve.

If we define a function g by

$$\theta \mapsto g(\theta) := J((\mathrm{Id} + \theta)(\Omega))$$

for θ in some suitable Hilbert space H, we want to compute the gradient $\nabla g(\theta)$ to find a direction of decent for our gradient algorithm. It is linked to the directional derivative $\partial g(\theta)(\psi)$, which is given by

$$\partial g(\theta)(\psi) = \mathrm{d}J((\mathrm{Id} + \theta)(\Omega))(\psi \circ (\mathrm{Id} + \theta)^{-1}), \tag{4.8}$$

as calculated in [10, Eq. (57)] for $(\mathrm{Id} + \theta)$ invertible. (4.8) is also the link to our shape derivative. The gradient $\nabla^H g(\theta)$ is thus defined as

$$\partial g(\theta)(\psi) := \left(\nabla^H g(\theta), \psi\right)_H \quad \text{for all } \psi \in H. \tag{4.9}$$

There are various different choices for H and we will compare some of them in our further experiments. The shape optimization algorithm is given in Algorithm 1.

As for the implementation in NGSolve $\mathrm{d}J(\Omega_n)$ is a linear form which takes a (vector valued) test function as input. This way it can be used as the left-hand side of (4.9) and calculating $\nabla^h g(\theta_n)$ is achieved by solving the problem: Find $\nabla^h g(\theta_n) \in H$ such that $a_H(\nabla^H g(\theta_n), \psi) = \mathrm{d}J(\Omega_n)(\psi)$ for all $\psi \in H$, where the bilinear form $a_H(.,.)$ depends on the chosen Hilbert space H.

The code for the following experiments can be found in Appendix A.4. Our FES is of order 3 and we choose a maximum element diameter of 0.02. For the input of Algorithm 1 we set Ω_0 as a circle of radius 0.3 centered at $M_2 = (0.5, 0.5)$ as our initial shape as it was done in [17]. The constants are given by $\gamma = 100$, $c = 1000$, $\alpha = 0.5$, $\mathrm{tol} = 10^{-7}$, $N_{max} = 100$, $\zeta = 10^{-4}$ and $\overline{\lambda} = \max(0, f_\Omega - \Delta\varphi)$.

As Hilbert spaces to calculate $\nabla^H g(\theta)$ we compare the ones proposed in [10]. They are given by

$$\begin{aligned}
H_1 &= H_0^1(D)^2, \quad (\theta, \psi)_{H_1} := \int_D \nabla\theta : \nabla\psi + \theta \cdot \psi \,\mathrm{d}x, \\
H_2 &= H_0^1(D)^2, \quad (\theta, \psi)_{H_2} := \int_D \varepsilon(\theta) : \varepsilon(\psi) + \theta \cdot \psi \,\mathrm{d}x, \\
H_3 &= H_0^1(D)^2, \quad (\theta, \psi)_{H_3} := \int_D \varepsilon(\theta) : \varepsilon(\psi) + \theta \cdot \psi + \gamma_{CR}\mathcal{B}\theta \cdot \mathcal{B}\psi \,\mathrm{d}x,
\end{aligned} \tag{4.10}$$

where $\varepsilon(\theta) := \frac{1}{2}(\nabla\theta + \nabla\theta^T)$, $\gamma_{CR} > 0$ and

$$\mathcal{B} := \begin{pmatrix} -\partial_x & \partial_y \\ \partial_y & \partial_x \end{pmatrix}.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithm 1 Gradient Algorithm

INPUT: Initial domain Ω_0, target function $\overline{y}$, obstacle φ, constants $\gamma, c, \alpha, \text{tol} > 0$, $N_{max} > 0$, $\zeta \geq 0$, arbitrary $\overline{\lambda} \in H^1(D)$
OUTPUT: Solution of (3.3)

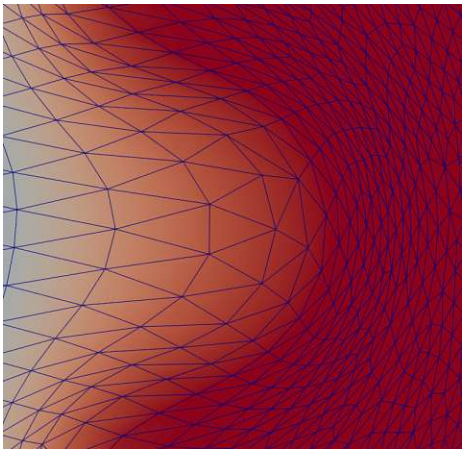
$n \leftarrow 0$
$\theta_0 \leftarrow 0$
do
  Calculate y_{γ,c,Ω_n} by solving (3.11)
  Calculate p_{γ,c,Ω_n} by solving (3.12)
  Calculate $\mathrm{d}J(\Omega_n)$ as given in (3.30)
  Calculate $\nabla^H g(\theta_n)$ by solving (4.9)
  while TRUE **do**
    if $J((\mathrm{Id} - \alpha\nabla^H g(\theta_n)(\Omega_n)) < J(\Omega_n) - \zeta\alpha\|\nabla^H g(\theta)\|^2$ **then**
      $\Omega_{n+1} \leftarrow (\mathrm{Id} - \alpha\nabla^H g(\theta_n))(\Omega_n)$
      $\theta_{n+1} \leftarrow \theta_n - \alpha\nabla^H g(\theta_n)$
      $n \leftarrow n + 1$
      Increase α
      Break inner while-loop
    else
      Decrease α
    end if
  end while
while $n < N_{max}$ and $\|\nabla g(\Omega_n)\| > \text{tol}$

RETURN: Ω_n.

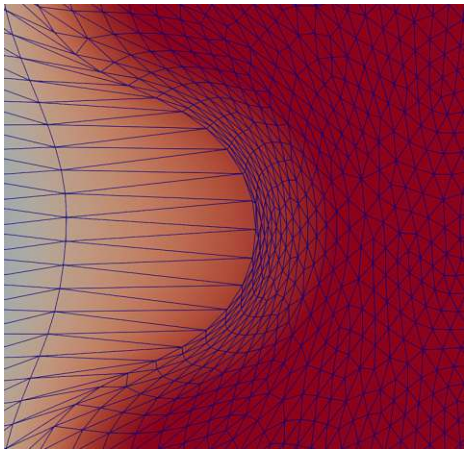
The additional term $\gamma_{CR}\mathcal{B}\theta \cdot \mathcal{B}\psi$ in the last inner product makes the gradient $\nabla^{H_3} g(\theta)$ almost conformal, which should preserve the quality of the mesh.

In Figure 4.7 we see the value of $J(\Omega_n)$ over the first 100 iterations and the various Hilbert spaces for both the Integral formula (3.30) and NGSolve's DiffShape(). The alternative stopping criteria $\|\nabla g(\Omega_n)\| \leq \text{tol}$ is never reached. After a first quick descent convergence slows down in the cases of H_1 and H_2. In our example H_3 performs worse then the other two Hilbert spaces in the beginning. At some point in the case of the Integral formula and H_2 the program always runs into to the **else** condition of the algorithm until $\alpha = 0$. This indicates the need of an additional stopping criteria. In the case of H_3 NGSolve's Diffshape() seems to perform slightly better. In Figure 4.8 finally we plot $\|\nabla^H g(\theta_n)\|_{L^2(D)}$ for for each iteration which also converges to 0.

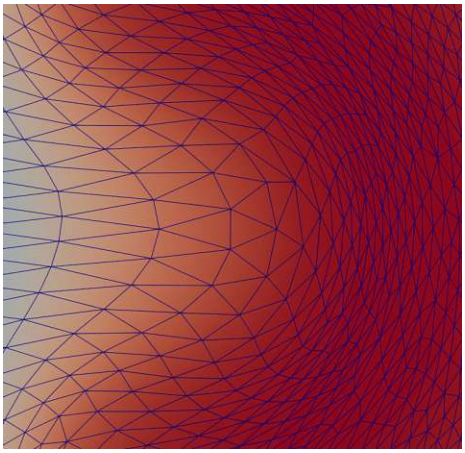
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



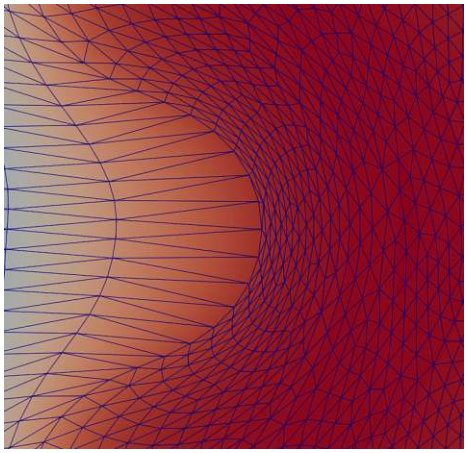
(a) H_1 using (3.30)



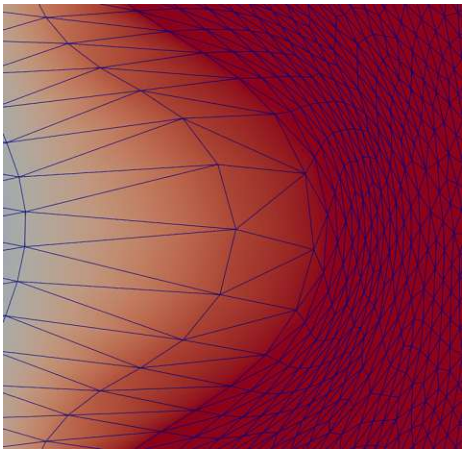
(b) H_1 using NGSolve's DiffShape()



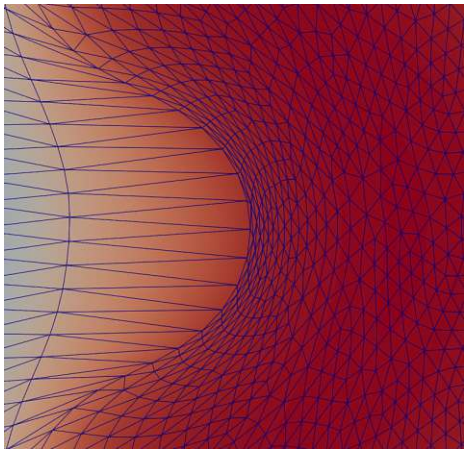
(c) H_2 using (3.30)



(d) H_2 using NGSolve's DiffShape()



(e) H_3 using (3.30)



(f) H_3 using NGSolve's DiffShape()

Figure 4.6.: Detail of the mesh for various Hilbert spaces. The meshes using H_3 become unusable at some point and would need to be optimized regularly in our example.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

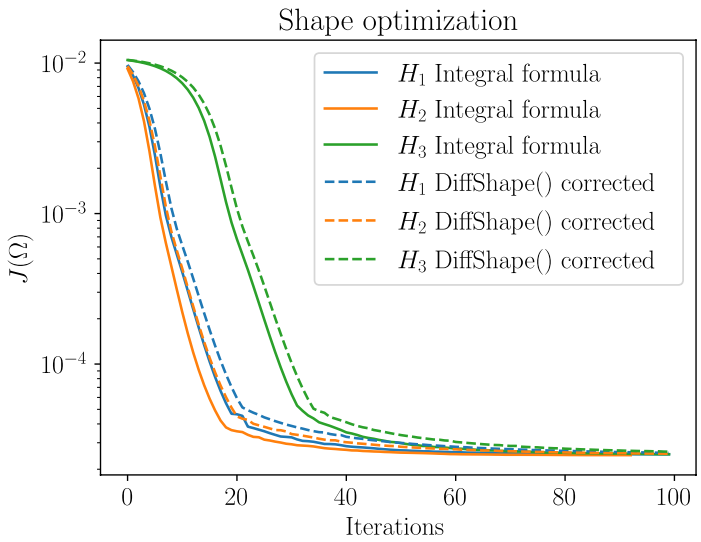


Figure 4.7.: The value of $J(\Omega)$ as the iterations in the gradient algorithm progress for various Hilbert spaces. The integral formula (3.30) performs better in the beginning for H_1 and H_2 but ends in a loop at some point. H_3 does not perform as well as the 'simpler' Hilbert spaces.

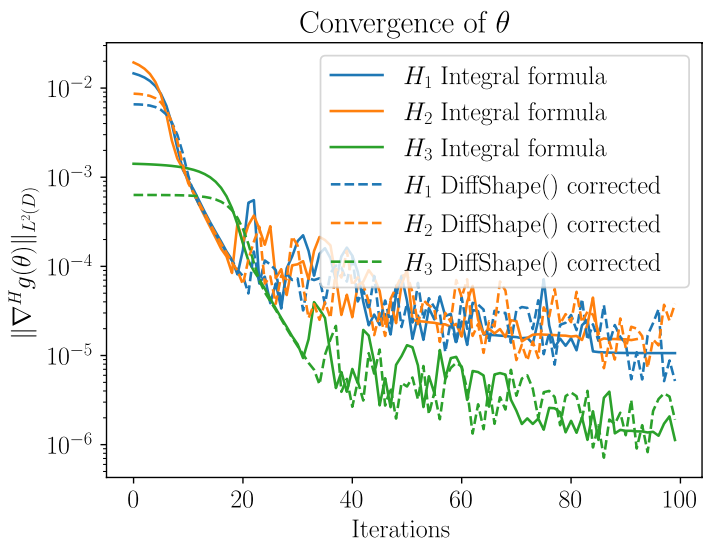


Figure 4.8.: The value of $\|\nabla^H g(\theta_n)\|_{L^2(D)}$ as the iterations in the gradient algorithm progress for various Hilbert spaces. See also Figure 4.7 for the convergence of the shape function.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5. The Unregularized Obstacle Problem

In this chapter we want to solve our shape optimization problem from Section 3.1 in a different way. The difference is that this time we do not perform any regularization of our problem. We look at the same problem as described in Section 3.1 with the additional assumptions of $(e)_i = 0$ for $i = 1, ..., d$ and $b \geq 0$. Furthermore throughout this chapter we will consider the case of a dynamic obstacle φ_Ω depending on Ω. This chapter largely follows [11] on which all the theorems and proofs in this chapter are based on, but we allow for a more general bilinear form $a(.,.)$. In [11] the bilinear form has the form

$$a(y, v) := \int_D \nabla y^T \nabla v + \lambda y v \, \mathrm{d}x$$

for some constant $\lambda > 0$, whereas we allow for any bilinear form of the form

$$a(y, v) := \int_D \nabla y^T M(x) \nabla v + b(x) y v \, \mathrm{d}x$$

with M and b satisfying the conditions stated in Section 3.1 and $b(x) \geq 0$ for all $x \in D$.

The outline of this chapter is as follows. We first review some abstract theorems in operator theory. We then take our specific problem and show how and why we can apply the theory proven previously. We conclude this chapter by giving an alternative shape derivative formula which does not require any previous regularization of the problem.

5.1. Abstract Operator Theory

We start with a few definitions.

Definition 5.1. Let $\mathcal{K} \subseteq V$ be a subset of a real Banach space V. The *radial cone* at $y \in \mathcal{K}$ of the set $\mathcal{K}$ is defined by

$$C_y(\mathcal{K}) := \{w \in V : \exists t > 0 : y + tw \in \mathcal{K}\}.$$

Definition 5.2. Let $\mathcal{K} \subseteq V$ be a subset of a real Banach space V. The *tangent cone* at $y \in \mathcal{K}$ of the set $\mathcal{K}$ is defined by the closure of $C_y(\mathcal{K})$ in V

$$T_y(\mathcal{K}) := \overline{C_y(\mathcal{K})}^V.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Definition 5.3. Let $\mathcal{K} \subseteq V$ be a subset of a real Banach space V and V' its dual space. The *normal cone* at $y \in \mathcal{K}$ of the set $\mathcal{K}$ is defined by

$$N_y(\mathcal{K}) := \{w' \in V' : \forall v \in \mathcal{K} \text{ it holds } \langle w', v - y\rangle_V \leq 0\}. \tag{5.1}$$

Definition 5.4. Let V be a real Banach space and V' its dual space. The *orthogonal complements* $[y]^\perp$ and $[y']^\perp$ of elements $y \in V$ and $y' \in V'$ are defined by

$$[y]^\perp := \{w' \in V' : \langle w', y\rangle_V = 0\}$$
$$[y']^\perp := \operatorname{kern}(y') := \{w \in V : \langle y', w\rangle_V = 0\}$$

Definition 5.5. Let $\mathcal{K} \subseteq V$ be a closed convex subset of a real Banach space V. We say $\mathcal{K}$ is *polyhedric* if

$$\forall y \in \mathcal{K}, \forall w \in N_y(\mathcal{K}) : \overline{C_y(\mathcal{K}) \cap [w]^\perp}^V = T_y(\mathcal{K}) \cap [w]^\perp.$$

The definition of polyhedric is rather abstract in this context. Important for us is that our upper obstacle sets $\mathcal{K}_{\varphi_\Omega}$ are polyhedric. The proof for this claim can be found in [11, Appendix A].

With these definitions we can now prove our first theorem in this chapter.

Theorem 5.6. *Let $\mathcal{K} \subseteq V$ be a polyhedric subset of a Banach space V. Let $\mathbf{A} : \mathcal{K} \to V'$ be an operator and let $y \in \mathcal{K}$ be a solution of the inequality*

$$\langle \mathbf{A}(y), v - y\rangle_V \geq 0 \qquad \forall v \in \mathcal{K}.$$

Then it holds that

$$\overline{C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))} = T_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y)).$$

Proof. By Definition 5.3 of the normal cone we obtain that $-\mathbf{A}(y) \in N_y(\mathcal{K})$. Since $\mathcal{K}$ is polyhedric we thus get by definition

$$\overline{C_y(\mathcal{K}) \cap [-\mathbf{A}(y)]^\perp}^V = T_y(\mathcal{K}) \cap [-\mathbf{A}(y)]^\perp,$$

which by definition of the orthogonal complement is equivalent to

$$\overline{C_y(\mathcal{K}) \cap \operatorname{kern}(-\mathbf{A}(y))}^V = T_y(\mathcal{K}) \cap \operatorname{kern}(-\mathbf{A}(y)).$$

Since $\operatorname{kern}(-\mathbf{A}(y)) = \operatorname{kern}(\mathbf{A}(y))$ by linearity we have shown the theorem. □

In this section we require our problem to fulfill two assumptions. The first one is the following.

Assumption 5.7. *Let $\tau > 0$. Assume that $(\mathbf{A}_t) : \mathcal{K} \to V'$, $t \in [0, \tau]$ is a family of operators satisfying the following:*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

(i) There exists an $\alpha > 0$ such that for all $t \in [0, \tau]$ and for all $y, v \in \mathcal{K}$ it holds that

$$\alpha \|y - v\|_V^2 \leq \langle \mathbf{A}_t(y) - \mathbf{A}_t(v), y - v \rangle_V. \tag{5.2}$$

(ii) For every $y \in \mathcal{K}$ there exists some $c > 0$ such that for all $t \in [0, \tau]$ and for all $v \in \mathcal{K}$ it holds that

$$|\langle \mathbf{A}_t(y) - \mathbf{A}_0(y), y - v \rangle_V \leq ct \|y - v\|_V. \tag{5.3}$$

Using this assumption we can prove the following Lipschitz continuity like result.

Theorem 5.8. *Let $(\mathbf{A}_t) : \mathcal{K} \to V'$ be a family of operators satisfying Assumption 5.7. For every $t \in [0, \tau]$ let $y^t \in \mathcal{K}$ be a solution of the variational inequality: Find $y^t \in \mathcal{K}$ such that for all $v \in \mathcal{K}$ it holds that*

$$\langle \mathbf{A}_t(y^t), v - y^t \rangle_V \geq 0. \tag{5.4}$$

Then there exists a constant $c > 0$ such that for all $t \in [0, \tau]$ it holds that

$$\|y^t - y^0\|_V \leq ct.$$

Proof. Using (5.2), using (5.4) twice in a row and using (5.3) we obtain

$$\begin{aligned}
\alpha \|y^t - y^0\|_V^2 &\leq \langle \mathbf{A}_t(y^t) - \mathbf{A}_t(y^0), y^t - y^0 \rangle_V \\
&\leq -\langle \mathbf{A}_t(y^0), y^t - y^0 \rangle_V \\
&= \langle \mathbf{A}_0(y^0), y^t - y^0 \rangle_V + \langle \mathbf{A}_0(y^0) - \mathbf{A}_t(y^0), y^t - y^0 \rangle_V \\
&\leq \langle \mathbf{A}_0(y^0) - \mathbf{A}_t(y^0), y^t - y^0 \rangle_V \\
&\leq ct \|y^t - y^0\|_V.
\end{aligned}$$

Dividing both sides by $\alpha > 0$ and $\|y^t - y^0\|_V$ concludes the proof. □

In the following let $\tau > 0$. We define for $t \in [0, \tau]$

$$Y(t) := \{y^t \in \mathcal{K} : \forall v \in \mathcal{K}, \langle \mathbf{A}_t(y^t), v - y^t \rangle_V \geq 0\} \tag{5.5}$$

and denote $y := y^0$ and $\mathbf{A} := \mathbf{A}_0$.

Assumption 5.9. *Let $\mathcal{K} \subseteq V$ be a polyhedric subset and let $(\mathbf{A}_t) : \mathcal{K} \to V'$ be a family of operators satisfying the following:*

(i) For all $v, w \in V$ and all $y \in \mathcal{K}$ it holds that

$$\langle \partial \mathbf{A}(y) w, v \rangle_V := \lim_{t \to 0^+} \left\langle \frac{\mathbf{A}(y + tw) - \mathbf{A}(y)}{t}, v \right\rangle_V$$

and

$$\langle \mathbf{A}'(y), v \rangle_V := \lim_{t \to 0^+} \left\langle \frac{\mathbf{A}_t(y) - \mathbf{A}(y)}{t}, v \right\rangle_V$$

exist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

(ii) *For all sequences* $(t_n)_{n\in\mathbb{N}}$ *with* $\lim_{n\to\infty} t_n = 0$, *for all sequences* $(v_n)_{n\in\mathbb{N}}$ *in* V *converging weakly to some* $v \in V$ *and for all* $y^{t_n} \in Y(t_n)$ *converging strongly to some* $y \in \mathcal{K}$ *where* $Y(t_n)$ *is defined by* (5.5), *it holds that*

$$\langle \mathbf{A}'(y), v\rangle_V = \lim_{n\to\infty} \left\langle \frac{\mathbf{A}_{t_n}(y^{t_n}) - \mathbf{A}(y^{t_n})}{t_n}, v_n \right\rangle_V$$

(iii) *There exists a sequence* $(t_n)_{n\in\mathbb{N}}$ *with* $\lim_{n\to\infty} t_n = 0$ *such that* $y^{t_n} \in Y(t_n)$ *converges strongly to* $y \in \mathcal{K}$ *and* $(y_{t_n} - y)/t_n$ *converges weakly to some* $z \in V$ *and*

$$\langle \partial\mathbf{A}(y)z, z\rangle_V \le \liminf_{n\to\infty} \left\langle \frac{\mathbf{A}(y^{t_n}) - \mathbf{A}(y)}{t_n}, \frac{y^{t_n} - y}{t_n} \right\rangle_V$$

and for all sequences $(v_n)_{n\in\mathbb{N}}$ *in* V *converging strongly to* $v \in V$

$$\langle \partial\mathbf{A}(y)z, v\rangle_V = \lim_{n\to\infty} \left\langle \frac{\mathbf{A}(y^{t_n}) - \mathbf{A}(y)}{t_n}, v_n \right\rangle_V .$$

The following result is the main result in this section. Later this result gives us a characterization for the material derivative $\dot{y}_\theta$ of our state equation, which we need to give an alternative shape derivative formula.

Theorem 5.10. *Let* V *be a reflexive Banach space and* $\mathcal{K} \subseteq V$ *a polyhedric subset. Let* $\tau > 0$ *and* $\mathbf{A}_t : \mathcal{K} \to V'$ *for* $t \in [0, \tau]$ *be a family of operators satifying Assumption 5.7 and 5.9. Let* $y^t \in Y(t)$.

Then the material derivative $\dot{y} := \text{weak-}\lim_{t\to 0^+} (y^t - y)/t$ *exists and satisifies*

$$\dot{y} \in T_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y)) \text{ and} \tag{5.6}$$

$$\forall v \in T_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y)) : \quad \langle \partial\mathbf{A}(y)\dot{y}, v - \dot{y}\rangle_V \ge -\langle \mathbf{A}'(y), v - \dot{y}\rangle_V . \tag{5.7}$$

Proof. Since $y^t \in Y(t)$ for $t \in [0, \tau]$ we have by definition of $Y(t)$

$$\forall v \in \mathcal{K} : \langle \mathbf{A}_t(y^t), v - y^t\rangle_V \ge 0 \text{ and } \langle \mathbf{A}(y), v - y\rangle_V \ge 0.$$

We now test the first equality with $\varphi = y$ and the second one with $\varphi = y^t$ and have

$$\langle \mathbf{A}_t(y^t), y - y^t\rangle_V \ge 0 \text{ and } \langle \mathbf{A}(y), y^t - y\rangle_V \ge 0.$$

Dividing both equations by $t > 0$ and setting $z^t := (y^t - y)/t$ we get

$$\langle \mathbf{A}_t(y^t), z^t\rangle_V \le 0 \text{ and } \langle \mathbf{A}(y), z^t\rangle_V \ge 0. \tag{5.8}$$

We can now apply Theorem 5.8 and obtain that $y^t \to y$ strongly in V and that z^t is bounded in V. Thus we can find a weakly convergent subsequence z^{t_n} which converges to $\dot{y} \in V$ as $n \to \infty$. We now take a look at

$$\begin{aligned}
&\langle \mathbf{A}_{t_n}(y^{t_n}), z^{t_n}\rangle_V - \langle \mathbf{A}(y), \dot{y}\rangle_V \\
&\quad = \langle \mathbf{A}_{t_n}(y^{t_n}) - \mathbf{A}(y^{t_n}), z^{t_n}\rangle_V + \left\langle \frac{\mathbf{A}(y^{t_n}) - \mathbf{A}(y)}{t_n}, y^{t_n} - y \right\rangle_V + \langle \mathbf{A}(y), z^{t_n} - \dot{y}\rangle_V .
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

The first term on the right-hand side converges to 0 by Assumption 5.7 ii). The second term converges to 0 due to Assumption 5.9 iii) and i). Note that $y^{t_n} - y$ converges strongly and thus the limit exists by our assumptions. The third term finally converges to 0 by definition of weak convergence. Therefore

$$\langle \mathbf{A}_{t_n}(y^{t_n}), z^{t_n} \rangle_V \to \langle \mathbf{A}(y), \dot{y} \rangle_V$$

as $n \to \infty$ and with (5.8) we thus have $0 \leq \langle \mathbf{A}(y), \dot{y} \rangle_V \leq 0$. Therefore $\dot{y} \in \operatorname{kern}(\mathbf{A}(y))$.

By definition of z^{t_n} and the radial cone we have $z^{t_n} \in C_y(\mathcal{K})$. Since $z^{t_n} \rightharpoonup \dot{y}$ weakly and $\mathcal{K}$ is polyhedric and thus convex we can apply Mazur's lemma which states that for a Banach space V and a sequence (z^{t_n}) in V converging weakly to some $\dot{y} \in V$ there exists a function $N : \mathbb{N} \to \mathbb{N}$ and a sequence of sets of real numbers $\{\beta(n)_k : k = n, ..., N(n)\}$ such that $\beta(n)_k \geq 0$, $\sum_{k=n}^{N(n)} \beta(n)_k = 1$ and such that the sequence $(v_n)_{n\in\mathbb{N}} := \sum_{k=n}^{N(n)} \beta(n)_k z^{t_k}$ (which is a convex combination) converges strongly in V to $\dot{y}$. In our case this implies $\dot{y} \in \overline{C_y(\mathcal{K})}^V = T_y(\mathcal{K})$ and finishes the proof of (5.6).

For (5.7) we have again due to $y^t \in Y(t)$ for all $v \in \mathcal{K}$

$$\begin{aligned}\left\langle \mathbf{A}(y^t) - \mathbf{A}(y), v - y^t \right\rangle_V &= \left\langle \mathbf{A}(y^t) - \mathbf{A}_t(y^t), v - y^t \right\rangle_V + \left\langle \mathbf{A}_t(y^t) - \mathbf{A}(y), v - y^t \right\rangle_V \\ &\geq \left\langle \mathbf{A}(y^t) - \mathbf{A}_t(y^t), v - y^t \right\rangle_V - \left\langle \mathbf{A}(y), v - y^t \right\rangle_V .\end{aligned}$$

We now plug a certain function as a test function into this equation. By definition of the radial cone $C_y(\mathcal{K})$, $y \in \mathcal{K}$ and convexity of $\mathcal{K}$ for every $v \in C_y(\mathcal{K})$ there exists a constant $t^* > 0$ such that for all $t \in [0, t^*]$ we have $y + tv \in \mathcal{K}$. Plugging this $y + tv$ into above equation we get

$$\begin{aligned}&\left\langle \mathbf{A}(y^t) - \mathbf{A}(y), tv - (y^t - y) \right\rangle_V \\ &\qquad \geq \left\langle \mathbf{A}(y^t) - \mathbf{A}_t(y^t), tv - (y^t - y) \right\rangle_V - \left\langle \mathbf{A}(y), tv - (y^t - y) \right\rangle_V .\end{aligned}$$

We divide both sides by t^2 and get

$$\left\langle \frac{\mathbf{A}(y^t) - \mathbf{A}(y)}{t}, v - z^t \right\rangle_V \geq - \left\langle \frac{\mathbf{A}_t(y^t) - \mathbf{A}(y^t)}{t}, v - z^t \right\rangle_V - \frac{1}{t} \left\langle \mathbf{A}(y), v - z^t \right\rangle_V . \tag{5.9}$$

Now we take $v \in C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))$. For such a v we have $\langle \mathbf{A}(y), v \rangle_V = 0$ and thus the last term satisifes

$$\left\langle \mathbf{A}(y), v - z^t \right\rangle_V = \langle \mathbf{A}(y), v \rangle_V - \left\langle \mathbf{A}(y), \frac{y^t - y}{t} \right\rangle_V \leq 0$$

as $y^t \in \mathcal{K}$ and $y \in Y(0)$. We can therefore leave this term away in (5.9) and have

$$\left\langle \frac{\mathbf{A}(y^t) - \mathbf{A}(y)}{t}, v - z^t \right\rangle_V \geq - \left\langle \frac{\mathbf{A}_t(y^t) - \mathbf{A}(y^t)}{t}, v - z^t \right\rangle_V .$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

We now want to calculate the lim sup of both sides. For the left-hand side we get

$$\begin{aligned}\limsup_{t\to 0}\left\langle \frac{\mathbf{A}(y^t)-\mathbf{A}(y)}{t}, v-z^t\right\rangle_V &= \limsup_{t\to 0}\left\langle \frac{\mathbf{A}(y^t)-\mathbf{A}(y)}{t}, v\right\rangle_V \\ &\quad - \limsup_{t\to 0}\left\langle \frac{\mathbf{A}(y^t)-\mathbf{A}(y)}{t}, z^t\right\rangle_V \\ &\le \langle \partial\mathbf{A}(y)z, v\rangle_V - \langle \partial\mathbf{A}(y)z, z\rangle_V\end{aligned}$$

using Assumption 5.9 i) and iii) and the fact that $-\limsup(...) = \liminf -(...)$. For the right-hand side we do the same calculations and use Assumption 5.9 i) and ii) and thus get (5.7) for all $v \in C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))$. Since $C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))$ is dense in $\overline{C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))}^V$ we also have this inequality for $v \in \overline{C_y(\mathcal{K}) \cap \operatorname{kern}(\mathbf{A}(y))}^V$ and can finally apply Theorem 5.6 which concludes the proof. □

5.2. Application of the Operator Theory

In [11] the aim is to minimize an energy functional looking similar to this one

$$\mathcal{E}(\Omega, v) = \int_D \frac{1}{2}\nabla v^T M \nabla v + \frac{b}{2}v^2 + W_\Omega(x, v)\ \mathrm{d}x, \quad v \in H^1(D), \tag{5.10}$$

where $W_\Omega(x, .)$ is some given convex function. In this thesis we are specifically interested in the case $W_\Omega(x, v) = f_\Omega(x)v(x)$. Thus in the following we only consider this case and the theory in [11] simplifies accordingly.

Equations (5.10) and (3.1) are closely connected. Indeed the minimizer of this functional over the set $\mathcal{K}_{\varphi_\Omega}$ and the solution of our obstacle problem (3.1) are the same object if we chose $W_\Omega(x, v) = f_\Omega(x)v(x)$. We also note again that our obstacle $\varphi_\Omega \in H_0^1(D)$ might itself depend on Ω.

The following theorem links the energy functional with the solution of the obstacle problem.

Theorem 5.11. *The energy $\mathcal{E}$ in (5.10) admits for each Lipschitz domain $\Omega \subseteq D$ a unique minimum y_Ω on $\mathcal{K}_{\varphi_\Omega}$ which is given as the unique solution of the problem: Find $y_\Omega \in \mathcal{K}_{\varphi_\Omega}$ such that for all $v \in \mathcal{K}_{\varphi_\Omega}$ the following inequality holds true*

$$\int_D \nabla y_\Omega^T M \nabla(v - y_\Omega) + b y_\Omega(v - y_\Omega) + f_\Omega(x, y_\Omega)(v - y_\Omega)\ \mathrm{d}x \ge 0. \tag{5.11}$$

Proof. The proof uses standard arguments in calculus of variations so we are only going to sketch it here. We investigate the function

$$E : D \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R} : (x, z, p) \mapsto \frac{1}{2}p^T M p + \frac{b}{2}z^2 + f_\Omega z.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Since $b \geq 0$ the function E is convex in p. Furthermore there exist constants $\alpha, \beta \geq 0$ such that E can be bounded from below in the form

$$E(x, z, p) \geq \alpha |p|^2 - \beta,$$

since A is coercive and f_Ω bounded. Thus there exists a minimum $y_\Omega \in \mathcal{K}_{\varphi_\Omega}$ of $\mathcal{E}$.

As for the uniqueness of y_Ω we have to check that for all $x \in D$, $\lambda \in (0,1)$ and $(z_1, p_1), (z_2, p_2) \in \mathbb{R} \times \mathbb{R}^d$ with $p_1 \neq p_2$ it holds that

$$E(x, (1-\lambda)z_1 + \lambda z_2, (1-\lambda)p_1 + \lambda p_2) < (1-\lambda)E(x, z_1, p_1) + \lambda E(x, z_2, p_2).$$

That is true again due to coerciveness of M, $b \geq 0$ and convexity of $z \mapsto f_\Omega z$. □

The analysis in the remainder of this section is performed over the standard obstacle set

$$\mathcal{K}_0 := \{v \in H_0^1(D) : v \leq 0 \text{ a.e.}\}.$$

Therefore we also have to shift the state function y_Ω and the test functions v

$$\begin{aligned} \widetilde{y}_\Omega &:= y_\Omega - \varphi_\Omega \\ \widetilde{v} &:= v - \varphi_\Omega. \end{aligned}$$

Our variational inequality (5.11) becomes: Find $\widetilde{y}_\Omega \in \mathcal{K}_0$ such that for all $v \in \mathcal{K}_0$ the following inequality holds true

$$\begin{aligned} &\int_D \nabla \widetilde{y}_\Omega^T M \nabla (v - \widetilde{y}_\Omega) + b y_\Omega (v - \widetilde{y}_\Omega) + f_\Omega(x, \widetilde{y}_\Omega + \varphi_\Omega)(v - \widetilde{y}_\Omega) \, \mathrm{d}x \\ &\qquad \geq -\int_D \nabla \varphi_\Omega^T M \nabla (v - \widetilde{y}_\Omega) + b \varphi_\Omega (v - \widetilde{y}_\Omega) \, \mathrm{d}x. \end{aligned} \tag{5.12}$$

We also introduce for $\theta \in C_c^1(\overline{D}, \mathbb{R}^d)$ the following notation.

$$\begin{aligned} f_{\Omega_t}^t &:= f_{\Omega_t} \circ \Phi_t^\theta, \\ \varphi_{\Omega_t}^t &:= \varphi_{\Omega_t} \circ \Phi_t^\theta \\ \xi(t) &:= \det(\partial \Phi_t^\theta) \\ M(t) &:= \xi(t)(\partial \Phi_t^\theta)^{-1}(M \circ \Phi_t^\theta)(\partial \Phi_t^\theta)^{-T} \\ y_{\Omega_t}^t &:= y_{\Omega_t} \circ \Phi_t^\theta. \end{aligned} \tag{5.13}$$

Next we recall the Sobolev exponent 2^* defined by

$$2^* := \begin{cases} \frac{2d}{d-2} & \text{if } d > 2, \\ \text{arbitrary in } [0, +\infty) & \text{if } d = 2, \\ +\infty & \text{if } d = 1 \end{cases}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and its conjugate $(2^*)'$ by

$$(2^*)' := \begin{cases} \frac{2^*}{2^*-1} & \text{if } 2^* < +\infty, \\ 1 & \text{if } 2^* = +\infty. \end{cases}$$

We make use of it later when we use the Sobolev embedding $H^1(D) \hookrightarrow L^{2^*}(D)$. It also shows up in the following two assumptions.

Assumption 5.12.

(i) For all $\theta \in C_c^1(D, \mathbb{R}^d)$ there exists a constant $c > 0$ such that for all $t \in [0, \tau]$ and all $v \in H^1(D)$

$$\|f_{\Omega_t}^t - f_\Omega\|_{L^{(2^*)'}(D)} \leq ct.$$

(ii) For all $\theta \in C_c^1(D, \mathbb{R}^d)$ there exists a constant $c > 0$ such that for all $t \in [0, \tau]$

$$\|\varphi_{\Omega_t}^t - \varphi_\Omega\|_{H^1(D)} \leq ct.$$

We define the bilinear form $a_t(.,.)$ by

$$a_t(v, z) := \int_D \left((\partial\Phi_t^\theta)^{-1}\nabla v^T (M \circ \Phi_t^\theta)(\partial\Phi_t^\theta)^{-1}\nabla z + (b \circ \Phi_t^\theta) vz\right) \det(\partial\Phi_t^\theta)\, \mathrm{d}x,$$

the operator $\mathbf{A}_t : \mathcal{K}_{\varphi_\Omega} \to H^1(D)'$ by

$$\langle \mathbf{A}_t(v), z\rangle_{H^1(D)} := a_t(v, z) + \int_D \xi(t) f_{\Omega_t}^t \, \mathrm{d}x$$

and the shifted operator $\widetilde{\mathbf{A}}_t : \mathcal{K}_0 \to H^1(D)'$ by

$$\widetilde{\mathbf{A}}_t(v) := \mathbf{A}_t(v + \varphi_{\Omega_t}^t). \tag{5.14}$$

Theorem 5.13. *Let Assumption 5.12 be satisified. Then the family of operators defined by* (5.14) *fulfills the properties*

(i) There exists an $\alpha > 0$ and a $t^* > 0$ such that for all $t \in [0, t^*]$ and for all $v, z \in \mathcal{K}_0$

$$\alpha\|v - z\|_{H^1(D)}^2 \leq \left\langle \widetilde{\mathbf{A}}_t(v) - \widetilde{\mathbf{A}}_t(z), v - z \right\rangle_{H_0^1(D)}.$$

(ii) For all $v \in \mathcal{K}_0$ there exists a constant $c > 0$ and a constant $t^* > 0$ such that for all $t \in [0, t^*]$ and for all $z \in \mathcal{K}_0$

$$\left| \left\langle \widetilde{\mathbf{A}}_t(v) - \widetilde{\mathbf{A}}(v), v - z \right\rangle_{H_0^1(D)} \right| \leq ct\|v - z\|_{H^1(D)}.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. Ad i). We have due to coercivity of $a(.,.)$

$$\|v-z\|^2_{H^1(D)} \leq \frac{1}{\mu} a(v-z, v-z) = \frac{1}{\mu} \int_D \nabla(v-z)^T M \nabla(v-z) + b(v-z)^2 \, \mathrm{d}x. \tag{5.15}$$

Let $v \neq z$. Then the right-hand side of this equation is strictly larger than 0. Since $M(t) \to M$ and $\xi(t) \to 1$ as $t \to 0$ due to Theorem 2.17 there exists a $t^* > 0$ such that for all $t \in [0, t^*]$

$$\int_D \nabla(v-z)^T M(t) \nabla(v-z) + \xi(t) b(v-z)^2 \, \mathrm{d}x > 0$$

and

$$\int_D \nabla(v-z)^T (M - M(t)) \nabla(v-z) + (1-\xi(t)) b(v-z)^2 \, \mathrm{d}x \\ < \frac{1}{2} \int_D \nabla(v-z)^T M \nabla(v-z) + b(v-z)^2 \, \mathrm{d}x.$$

From those two inequalities it follows that we can further estimate inequality (5.15) by

$$\begin{aligned} \frac{1}{\mu} \int_D \nabla(v-z)^T M \nabla(v-z) + b(v-z)^2 \, \mathrm{d}x \\ &\leq \frac{2}{\mu} \int_D \nabla(v-z)^T M(t) \nabla(v-z) + \xi(t) b(v-z)^2 \, \mathrm{d}x \\ &\leq \frac{2}{\mu} a_t(v-z, v-z) \\ &= \frac{2}{\mu} \left\langle \widetilde{\mathbf{A}}_t(v) - \widetilde{\mathbf{A}}_t(z), v-z \right\rangle_{H^1_0(D)}, \end{aligned}$$

since the term $\int_D \xi(t) f^t_{\Omega_t} \, \mathrm{d}x$ in the definition of $\left\langle \widetilde{\mathbf{A}}_t(v) - \widetilde{\mathbf{A}}_t(z), v-z \right\rangle_{H^1_0(D)}$ cancels. This concludes the first part of the proof.

Ad ii). We start by calculating

$$\begin{aligned} \left\langle \widetilde{\mathbf{A}}_t(v) - \widetilde{\mathbf{A}}(v), v-z \right\rangle_{H^1_0(D)} &= \left\langle \mathbf{A}_t(v + \varphi^t_{\Omega_t}) - \mathbf{A}(v + \varphi_\Omega), v-z \right\rangle_{H^1_0(D)} \\ &= a_t(v + \varphi^t_{\Omega_t}, v-z) + \int_D \xi(t) f^t_{\Omega_t}(v-z) \, \mathrm{d}x \\ &\quad - a(v + \varphi_\Omega, v-z) - \int_D f_\Omega(v-z) \, \mathrm{d}x \end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$
\begin{aligned}
&= \int_D \nabla(v+\varphi^t_{\Omega_t})^T M(t)\nabla(v-z) \\
&\qquad\qquad + \xi(t) b(v+\varphi^t_{\Omega_t})(v-z)\ \mathrm{d}x \\
&\quad - \int_D \nabla(v+\varphi_\Omega)^T M\nabla(v-z) + b(v+\varphi_\Omega)(v-z)\ \mathrm{d}x \\
&\quad + \int_D \xi(t) f^t_{\Omega_t}(v-z)\ \mathrm{d}x \\
&\quad - \int_D f_\Omega(v-z)\ \mathrm{d}x \\
&= \int_D \nabla v^T (M(t)-M)\nabla(v-z) \\
&\quad + \int_D \left((\nabla\varphi^t_{\Omega_t})^T M(t) - (\nabla\varphi_\Omega)^T M\right)\nabla(v-z)\ \mathrm{d}x \\
&\quad + \int_D b(\xi(t)-1)v(v-z)\ \mathrm{d}x \\
&\quad + \int_D b(\xi(t)\varphi^t_{\Omega_t} - \varphi_\Omega)(v-z)\ \mathrm{d}x \\
&\quad + \int_D \xi(t) f^t_{\Omega_t}(v-z)\ \mathrm{d}x \\
&\quad - \int_D f_\Omega(v-z)\ \mathrm{d}x.
\end{aligned}
$$

We now look at each term after the last equality individually. For the first term we have due to the convergence of the differential quotient in Theorem 2.17, $\|\nabla v\|_{L^2(D)} < +\infty$ and $\|\nabla(v-z)\|_{L^2(D)} \leq \|\nabla(v-z)\|_{H^1(D)}$ for some constant $c > 0$

$$
\begin{aligned}
\int_D \nabla v^T (M(t)-M)\nabla(v-z) &\leq \|M(t)-M\|_{L^\infty(D)}\|\nabla v\|_{L^2(D)}\|\nabla(v-z)\|_{L^2(D)} \\
&\leq ct\|v-z\|_{H^1(D)}.
\end{aligned}
$$

For the second term we have for the same reasons as in the first term in addition to Assumption 5.12 ii) for some other contant $c > 0$

$$
\begin{aligned}
&\int_D \left((\nabla\ \varphi^t_{\Omega_t})^T M(t) - (\nabla\varphi_\Omega)^T M\right)\nabla(v-z)\ \mathrm{d}x \\
&\qquad = \int_D \left((\nabla\varphi^t_{\Omega_t})^T M(t) - (\nabla\varphi^t_{\Omega_t})^T M + (\nabla\varphi^t_{\Omega_t})^T M - (\nabla\varphi_\Omega)^T M\right)\nabla(v-z)\ \mathrm{d}x \\
&\qquad \leq \|M(t)-M\|_{L^\infty(D)}\|(\nabla\varphi^t_{\Omega_t})^T\|\|\nabla(v-z)\|_{L^2(D)} \\
&\qquad\quad + \|M\|_{L^\infty}\|(\nabla\varphi^t_{\Omega_t})^T - (\nabla\varphi_\Omega)^T\|_{L^2(D)}\|\nabla(v-z)\|_{L^2(D)} \\
&\qquad \leq ct\|v-z\|_{H^1(D)}.
\end{aligned}
$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Analogously for the third term

$$\int_D b(\xi(t)-1)v(v-z)\,\mathrm{d}x \le \|\xi(t)-1\|_{L^\infty(D)}\|v\|_{L^2(D)}\|v-z\|_{L^2(D)}$$
$$\le ct\|v-z\|_{H^1(D)}$$

and the fourth term

$$\begin{aligned}
&\int_D b(\xi(t)\varphi^t_{\Omega_t}-\varphi_\Omega)(v-z)\,\mathrm{d}x\\
&\quad=\int_D b(\xi(t)\varphi^t_{\Omega_t}-\varphi^t_{\Omega_t}+\varphi^t_{\Omega_t}-\varphi_\Omega)(v-z)\,\mathrm{d}x\\
&\quad\le \|b(\xi(t)-1)\|_{L^\infty(D)}\|\varphi^t_{\Omega_t}\|_{L^2(D)}\|v-z\|_{L^2(D)}\\
&\qquad+\|b\|_{L^\infty(D)}\|\varphi^t_{\Omega_t}-\varphi_\Omega\|_{L^2(D)}\|v-z\|_{L^2(D)}\\
&\quad\le ct\|v-z\|_{H^1(D)}.
\end{aligned}$$

The last two terms we handle together. We add and subtract the same term and use Assumption 5.12 (ii) together with the same reasoning as before to obtain for some constant $c>0$

$$\begin{aligned}
&\int_D \xi(t)f^t_{\Omega_t}(v-z)\,\mathrm{d}x-\int_D f_\Omega(v-z)\,\mathrm{d}x\\
&\quad=\int_D\left(\xi(t)f^t_{\Omega_t}-f^t_{\Omega_t}+f^t_{\Omega_t}-f_\Omega\right)(v-z)\,\mathrm{d}x\\
&\quad\le\left(\|\xi(t)-1\|_{L^\infty(D)}\|f^t_{\Omega_t}\|_{L^{(2^*)'}(D)}+\|f^t_{\Omega_t}-f_\Omega\|_{L^{(2^*)'}(D)}\right)\|v-z\|_{L^{2^*}(D)}\\
&\quad\le ct\|v-z\|_{H^1(D)}.
\end{aligned}$$

Therefore we have $\left\langle \widetilde{\mathbf{A}}_t(v)-\widetilde{\mathbf{A}}(v), v-z\right\rangle_{H^1_0(D)}\le ct\|v-z\|_{H^1(D)}$ for some $c>0$ which finishes the proof. □

Theorem 5.14. *Let Assumptions 5.12 be satisified and let the family of operators* $(\widetilde{\mathbf{A}}_t)_t$ *be defined by* (5.14). *Let* $\widetilde{y}^t_{\Omega_t}$ *be the solution of the problem: Find* $\widetilde{y}^t_{\Omega_t}\in H^1_0(D)$ *such that for all* $v\in H^1_0(D)$

$$\left\langle \widetilde{\mathbf{A}}_t(\widetilde{y}^t_{\Omega_t}), v-\widetilde{y}^t_{\Omega_t}\right\rangle_{H^1_0(D)}\ge 0$$

holds true.

Then there exists $t^*>0$ *and a constant* $c>0$ *such that for all* $t\in[0,t^*]$

$$\|\widetilde{y}^t_{\Omega_t}-\widetilde{y}^0_\Omega\|_{H^1(D)}\le ct.$$

Proof. Theorem 5.13 shows that Assumption 5.7 is satisified. Therefore we can apply Theorem 5.8 which yields the result. □

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Assumption 5.15.

(i) For all $\theta \in C_c^1(D, \mathbb{R}^d)$ there exists a function $\dot{f}_\theta : \Omega \to \mathbb{R}$ such that we have $\dot{f}_\theta(., v) \in L^{(2^*)'}(D)$ and for all $t_n \to 0^+$ it holds that

$$\frac{f_{\Omega_{t_n}}^{t_n} - f_\Omega}{t_n} \to \dot{f}_\theta \quad \text{strongly in } L^{(2^*)'}(D) \text{ as } n \to +\infty.$$

(ii) For all $\theta \in C_c^1(D, \mathbb{R}^d)$ there exists a function $\dot{\varphi}_\theta \in H^1(D)$ such that

$$\frac{\varphi_{\Omega_{t_n}}^{t} - \varphi_\Omega}{t} \to \dot{\varphi}_\theta \quad \text{strongly in } H^1(D) \text{ as } t \to 0^+.$$

Theorem 5.16. *Let Assumptions 5.12 and 5.15 be satisified. Let $\widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0)$ be defined by*

$$\widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0) := T_{\widetilde{y}_\Omega}(\mathcal{K}_0) \cap \operatorname{kern}(\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)).$$

Then the weak material derivative $\dot{\widetilde{y}}_\theta$ of $t \mapsto \widetilde{y}_{\Omega_t}^t$ exists in all directions $\theta \in C_c^1(D, \mathbb{R}^d)$ and is characterized as the unique solution of the problem: Find $\dot{\widetilde{y}}_\theta \in \widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0)$ such that for all $v \in \widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0)$ it holds that

$$\left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{y}}_\theta, v - \dot{\widetilde{y}}_\theta \right\rangle_{H^1(D)} \geq -\left\langle \widetilde{\mathbf{A}}'(\widetilde{y}_\Omega), v - \dot{\widetilde{y}}_\theta \right\rangle_{H^1(D)}. \tag{5.16}$$

The functional derivatives $\partial\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{A}}'$ are given by

$$\left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{y}}_\theta, v \right\rangle_{H^1(D)} = a(\dot{\widetilde{y}}_\theta, v) \tag{5.17}$$

$$\begin{aligned}\left\langle \widetilde{\mathbf{A}}'(\widetilde{y}_\Omega), v \right\rangle_{H^1(D)} = &\int_D \nabla\widetilde{y}_\Omega^T M'(0)\nabla v + \xi'(0)\left(b\widetilde{y}_\Omega + f_\Omega\right) v \,\mathrm{d}x \\ &+ \int_D \dot{f}_\theta v \,\mathrm{d}x \\ &+ \int_D \nabla\varphi_\Omega^T M'(0)\nabla v + \xi'(0) b\varphi_\Omega v \,\mathrm{d}x \\ &+ \int_D \nabla(\dot{\varphi}_\theta)^T M\nabla v + b\dot{\varphi}_\theta v \,\mathrm{d}x\end{aligned} \tag{5.18}$$

Proof. We can apply Theorem 5.13, which gives us Assumption 5.7. If we can also show that Assumption 5.9 is satisified then we can apply Theorem 5.10 yielding our desired result. Thus we want to check (i) - (iii) in Assumption 5.9.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Ad (i). Given $u \in K$ and $v, w \in H^1(D)$ we calculate

$$\begin{aligned}
\left\langle \partial\widetilde{\mathbf{A}}(u)w, v \right\rangle_{H^1(D)} &:= \lim_{t\to 0^+} \left\langle \frac{\widetilde{\mathbf{A}}(u+tw) - \widetilde{\mathbf{A}}(u)}{t}, v \right\rangle_{H^1(D)} \\
&= \lim_{t\to 0^+} \left\langle \frac{\mathbf{A}(u+tw+\varphi_\Omega) - \mathbf{A}(u+\varphi_\Omega)}{t}, v \right\rangle_{H^1(D)} \\
&= \lim_{t\to 0^+} \left(\frac{1}{t}\left(a(u+tw+\varphi_\Omega, v) - a(u+\varphi_\Omega, v)\right) \right. \\
&\qquad \left. + \int_D \frac{1}{t}\left(f_\Omega - f_\Omega\right) v \,\mathrm{d}x \right) \\
&= a(w, v)
\end{aligned}$$

and

$$\begin{aligned}
\left\langle \widetilde{\mathbf{A}}'(u), v \right\rangle_{H^1(D)} &:= \lim_{t\to 0^+} \left\langle \frac{\widetilde{\mathbf{A}}_t(u) - \widetilde{\mathbf{A}}(u)}{t}, v \right\rangle_{H^1(D)} \\
&= \lim_{t\to 0^+} \left\langle \frac{\mathbf{A}_t(u+\varphi^t_{\Omega_t}) - \mathbf{A}(u+\varphi_\Omega)}{t}, v \right\rangle_{H^1(D)} \\
&= \lim_{t\to 0^+} \left(\frac{1}{t}\left(a_t(u+\varphi^t_{\Omega_t}, v) - a(u+\varphi_\Omega, v)\right) \right. \\
&\qquad \left. + \int_D \frac{1}{t}\left(\xi(t) f^t_{\Omega_t} - f_\Omega\right) v \,\mathrm{d}x \right).
\end{aligned}$$

Next we calculate the limit of the bilinear forms. We have

$$\begin{aligned}
&\frac{1}{t}\left(a_t(u+\varphi^t_{\Omega_t}, v) - a(u+\varphi_\Omega, v)\right) \\
&\quad = \frac{1}{t}\int_D \nabla(u+\varphi^t_{\Omega_t})^T M(t)\nabla v + \xi(t) b(u+\varphi^t_{\Omega_t}) v \\
&\qquad\qquad - \nabla(u+\varphi_\Omega)^T M \nabla v - b(u+\varphi_\Omega) v \,\mathrm{d}x \\
&\quad = \frac{1}{t}\int_D \nabla u^T (M(t) - M)\nabla v \,\mathrm{d}x + \frac{1}{t}\int_D \nabla(\varphi^t_{\Omega_t} - \varphi_\Omega)^T M(t)\nabla v \,\mathrm{d}x \\
&\qquad + \frac{1}{t}\int_D \nabla(\varphi_\Omega)^T (M(t) - M)\nabla v \,\mathrm{d}x + \frac{1}{t}\int_D (\xi(t) - 1) buv \,\mathrm{d}x \\
&\qquad + \frac{1}{t}\int_D \xi(t) b(\varphi^t_{\Omega_t} - \varphi_\Omega) v \,\mathrm{d}x + \frac{1}{t}\int_D (\xi(t) - 1) b\varphi_\Omega v \,\mathrm{d}x.
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Now taking the limit for each integral we get

$$\begin{aligned}\lim_{t\to 0^+}(\ldots) = &\int_D \nabla u^T M'(0)\nabla v \,\mathrm{d}x + \int_D \nabla(\dot\varphi_\theta)^T M\nabla v \,\mathrm{d}x \\ &+ \int_D \nabla(\varphi_\Omega)^T M'(0)\nabla v \,\mathrm{d}x + \int_D \xi'(0) buv \,\mathrm{d}x \\ &+ \int_D b\dot\varphi_\theta v \,\mathrm{d}x + \int_D \xi'(t) b\varphi_\Omega v \,\mathrm{d}x.\end{aligned}$$

For the limit of the other term we calculate

$$\begin{aligned}&\int_D \frac{1}{t}\left(\xi(t) f^t_{\Omega_t} - f_\Omega\right) v \,\mathrm{d}x \\ &\qquad \xrightarrow[t\to 0^+]{} \int_D \left(\xi'(0) f_\Omega + \dot f_\theta\right) v \,\mathrm{d}x.\end{aligned}$$

Ad (ii). We have to do the very same computations as in (i). Again everything works out due to our assumptions.

Ad (iii). We have

$$\begin{aligned}&\liminf_{n\to\infty}\left\langle \frac{\widetilde{\mathbf{A}}(\widetilde{y}^{t_n}_{\Omega_{t_n}}) - \widetilde{\mathbf{A}}(\widetilde{y}_\Omega)}{t_n}, \frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right\rangle_{H^1_0(D)} \\ &\quad = \liminf_{n\to\infty}\left\langle \frac{\mathbf{A}(\widetilde{y}^{t_n}_{\Omega_{t_n}} + \varphi_\Omega) - \mathbf{A}(\widetilde{y}_\Omega + \varphi_\Omega)}{t_n}, \frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right\rangle_{H^1_0(D)} \\ &\quad = \liminf_{n\to\infty} a\left(\frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}, \frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right) \\ &\quad = \liminf_{n\to\infty} \int_D \nabla\left(\frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right)^T M\nabla\left(\frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right) + b\left(\frac{\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega}{t_n}\right)^2 \mathrm{d}x.\end{aligned}$$

The last line is weakly lower semi-continuous and since $(\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega)/t_n \rightharpoonup z$ weakly we have by definition

$$\liminf_{n\to\infty}(\ldots) \geq \langle \partial\mathbf{A}(y)z, z\rangle_{H^1_0(D)}.$$

The second assumption is done analogously using weak convergence of $(\widetilde{y}^{t_n}_{\Omega_{t_n}} - \widetilde{y}_\Omega)/t_n$ again.

It remains to show that the solution $\dot{\widetilde{y}}_\theta \in \widetilde{S}_{\widetilde{y}}(\mathcal{K}_0)$ of our variational inequality (5.16) is indeed unique. Let us assume we have two solutions $\dot{\widetilde{y}}_\theta$ and $\dot{\widetilde{z}}_\theta$ of our variational inequality. We test both with the other solution and thus get

$$\begin{aligned}\left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{y}}_\theta, \dot{\widetilde{z}}_\theta - \dot{\widetilde{y}}_\theta\right\rangle_{H^1(D)} &\geq -\left\langle \widetilde{\mathbf{A}}'(\widetilde{y}_\Omega), \dot{\widetilde{z}}_\theta - \dot{\widetilde{y}}_\theta\right\rangle_{H^1(D)} \\ \left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{z}}_\theta, \dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta\right\rangle_{H^1(D)} &\geq -\left\langle \widetilde{\mathbf{A}}'(\widetilde{y}_\Omega), \dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta\right\rangle_{H^1(D)}.\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

By adding both terms and rearranging them we get

$$\left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{y}}_\theta - \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{z}}_\theta, \dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta \right\rangle_{H^1(D)} \leq 0.$$

Using (5.17) we get

$$\left\langle \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{y}}_\theta - \partial\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)\dot{\widetilde{z}}_\theta, \dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta \right\rangle_{H^1_0(D)} = a(\dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta, \dot{\widetilde{y}}_\theta - \dot{\widetilde{z}}_\theta) \leq 0,$$

which is a contradiction since we assume our bilinear form to be coercive and thus $\dot{\widetilde{y}}_\theta = \dot{\widetilde{z}}_\theta$. □

Theorem 5.17. *Under Assumptions 5.12 and 5.15 the material derivative $\dot{y}_\theta$ of solutions of the perturbed problem to (5.12) in direction $\theta \in C^1_c(D, \mathbb{R}^d)$ exists and is given as the solution of the problem: Find $\dot{y}_\theta \in S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega})$ such that for all $v \in S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega})$*

$$\begin{aligned} a(\dot{y}_\theta, v - \dot{y}_\theta) \geq &- \int_D \nabla u_\Omega^T M'(0)\nabla(v - \dot{y}_\theta) + \xi'(0)(bu_\Omega + f_\Omega)(v - \dot{y}_\theta)\ \mathrm{d}x \\ &- \int_D \dot{f}_\theta(v - \dot{y}_\theta)\ \mathrm{d}x, \end{aligned} \tag{5.19}$$

where $S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) := T_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) \cap \operatorname{kern}(\mathbf{A}(y_\Omega)) + \dot{\varphi}_\Omega$ and the function $M'(0)$ is defined as in Theorem 2.17.

Proof. We apply Theorem 5.16. Since $y_\Omega = \widetilde{y}_\Omega + \varphi_\Omega$ and $\dot{y}_\theta = \dot{\widetilde{y}}_\theta + \dot{\varphi}_\theta$ we find that $\dot{y}_\theta$ is the unique solution of the problem: Find $\dot{y}_\theta \in \widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0) + \dot{\varphi}_\theta$ such that for all $v \in \widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0) + \dot{\varphi}_\theta$ it holds that

$$\left\langle \partial\widetilde{\mathbf{A}}(y_\Omega - \varphi_\Omega)(\dot{y}_\theta - \dot{\varphi}_\theta), v - \dot{y}_\theta \right\rangle_{H^1(D)} \geq -\left\langle \widetilde{\mathbf{A}}'(y_\Omega - \varphi_\Omega), v - \dot{y}_\theta \right\rangle_{H^1(D)}.$$

Substituting the operators with equations (5.17) and (5.18) we get

$$\begin{aligned} &a(\dot{y}_\theta - \dot{\varphi}_\theta, v - \dot{y}_\theta) \\ &\quad \geq -\int_D \nabla(y_\Omega - \varphi_\Omega)^T M'(0)\nabla(v - \dot{y}_\theta) \\ &\qquad\qquad + \xi'(0)\left(b(y_\Omega - \varphi_\Omega) + f_\Omega\right)(v - \dot{y}_\theta)\ \mathrm{d}x \\ &\qquad - \int_D \dot{f}_\theta(v - \dot{y}_\theta)\ \mathrm{d}x - \int_D \nabla\varphi_\Omega^T M'(0)\nabla(v - \dot{y}_\theta) + \xi'(0)b\varphi_\Omega(v - \dot{y}_\theta)\ \mathrm{d}x \\ &\qquad - \int_D \nabla(\dot{\varphi}_\theta)^T M\nabla(v - \dot{y}_\theta) + b\dot{\varphi}_\theta(v - \dot{y}_\theta)\ \mathrm{d}x. \end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Some terms cancel and due to the definition of the bilinear form $a(.,.)$ this inequality simplifies to

$$a(\dot{y}_\theta, v - \dot{y}_\theta) \geq -\int_D \nabla(y_\Omega)^T M'(0)\nabla(v - \dot{y}_\theta) + \xi'(0)\,(by_\Omega + f_\Omega)\,(v - \dot{y}_\theta)\,\mathrm{d}x \\ - \int_D \dot{f}_\theta(v - \dot{y}_\theta)\,\mathrm{d}x,$$

which is exactly (5.19).

To conclude the proof we have to show that $S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) = \widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0) + \dot{\varphi}_\theta$. Due to Theorem 5.16 and the definition of $S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega})$ it is sufficient to show $T_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) \cap \operatorname{kern}(\mathbf{A}(y_\Omega)) = T_{\widetilde{y}_\Omega}(\mathcal{K}_0) \cap \operatorname{kern}(\widetilde{\mathbf{A}}(\widetilde{y}_\Omega))$.

By comparing the definitions of the tangent cones of $T_{y_\Omega}(\mathcal{K}_{\varphi_\Omega})$ and $T_{\widetilde{y}_\Omega}(\mathcal{K}_0)$ we immediately see $T_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) = T_{y_\Omega - \varphi_\Omega}(\mathcal{K}_0) = T_{\widetilde{y}_\Omega}(\mathcal{K}_0)$. As for the kernels of the operators we have $\operatorname{kern}(\widetilde{\mathbf{A}}(\widetilde{y}_\Omega)) = \operatorname{kern}(\mathbf{A}(\widetilde{y}_\Omega + \varphi_\Omega)) = \operatorname{kern}(\mathbf{A}(y_\Omega))$. □

Later on the following characterization is of great importance when we do our numerical experiments. It holds that

$$v \in S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega}) \Leftrightarrow \begin{cases} v \in H^1_0(D) \text{ with } v \leq \dot{\varphi}_\theta \text{ q.e. on } \{y_\Omega = \varphi_\Omega\} \\ \langle \mathbf{A}(y_\Omega), v - \dot{\varphi}_\theta \rangle_{H^1_0(D)} = 0, \end{cases} \tag{5.20}$$

where q.e. denotes quasi-everywhere, i.e. everywhere outside a polar set. To see why characterizations holds we need the following theorem, which is proven in [11, Appendix A].

Theorem 5.18. *Let $y \in \mathcal{K}_\varphi$. Let y^c denote the quasi-continuous representant of y. Then*

$$T_y(\mathcal{K}_\varphi) = \left\{ v \in H^1(D) : v^c \leq 0 \quad \text{q.e. on } \{y^c = v^c\} \right\}.$$

Here $\{y^c = v^c\} \subset \overline{D}$ denotes the set

$$\{y^c = v^c\} := \left\{ x \in \overline{D} : y^c(x) = v^c(x) \right\}.$$

The characterization now follows from the definitions of $S^\theta_{y_\Omega}(\mathcal{K}_{\varphi_\Omega})$ and of $\widetilde{S}_{\widetilde{y}_\Omega}(\mathcal{K}_0)$ together with Theorem 5.18.

We now have everything we need for an alternative shape derivative formula.

Theorem 5.19. *Let Assumptions 5.12 and 5.15 be satisfied and let $\Omega \subseteq D$ be a Lipschitz domain. Let $\theta \in C^1_c(D, \mathbb{R}^d)$ and $\Phi^\theta_t : D \to D$ with $\Phi^\theta_t(\Omega) = \Omega_t$ the corresponding flow. Suppose that for small $t > 0$ there is a Frechet differentiable function $\mathcal{J}$ with*

$$\mathcal{J} : C^{0,1}(D, \mathbb{R}^d) \times H^1(D) \to \mathbb{R} : (\Phi, y) \mapsto \mathcal{J}(\Phi, y)$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

such that for the the transported state $y^t_{\Omega_t} := y_{\Omega_t} \circ \Phi^\theta_t$ *with* y_{Ω_t} *being the unique solution of* (5.11) *on* Ω_t *we have*

$$J(\Omega_t) = \mathcal{J}(\Phi^\theta_t, y^t_{\Omega_t}).$$

Let $\dot{y}_\theta$ *be the unique solution of* (5.19).

Then the Eulerian semiderivative $\mathrm{d}J(\Omega)(\theta)$ *exists and is given by*

$$\mathrm{d}J(\Omega)(\theta) = \left\langle \mathrm{d}_\Phi \mathcal{J}(\mathrm{Id}, y^0_\Omega), \theta \right\rangle_{C^{0,1}(D;\mathbb{R}^d)} + \left\langle \mathrm{d}_y \mathcal{J}(\mathrm{Id}, y^0_\Omega), \dot{y}_\theta \right\rangle_{H^1(D)}. \tag{5.21}$$

Proof. Applying the chain rule on $\mathcal{J}$ together with Theorem 5.17 gives the result. □

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6. Numerical Experiments for the Unregularized Problem

In this chapter we apply the results of the previous chapter on the same problem as in Section 4.1. The numerical implementation poses more challenges this time as we have to solve the obstacle problem without regularization and we operate over the space $S^{\theta}_{y_\Omega}(\mathcal{K}_\varphi)$, which we need to grasp numerically. The majority of this chapter thus concerns our developement of an algorithm to overcome those challenges.

The outline of this chapter is as follows. First we specify which variational inequalities we have to solve to apply (5.21). We then explain how to solve the obstacle problem and how to obtain the material derivative for the unregularized problem. Next we state our shape optimization algorithm. In the last section we introduce another shape optimization approach which combines techniques from Chapter 3 with the techniques presented in this chapter. We conclude this chapter with a numerical comparison of the various techniques and some final remarks.

6.1. Experiment Setup

We want to solve the shape optimization problem (3.3) in the unregularized case. To do so we need the shape derivative $\mathrm{d}J(\Omega)(\theta)$ for a given vector field $\theta \in C^1_c(D, \mathbb{R}^2)$. Theorem 5.19 gives us an explicit formula which also outlines what problems we have to solve.

First there is the issue of getting $y^0_\Omega = y_\Omega$ where y_Ω solves the obstacle problem (5.11). In our specific case problem (5.11) simplifies to: Find $y_\Omega \in \mathcal{K}_\varphi$ such that for all $v \in \mathcal{K}_\varphi$ the following inequality holds true

$$\int_D \nabla y_\Omega^T \nabla(v - y_\Omega) - f_\Omega(v - y_\Omega)\ \mathrm{d}x \geq 0. \tag{6.1}$$

Second we have to know the material derivative $\dot{y}_\theta$, where $\dot{y}_\theta \in S^{\theta}_{y_\Omega}(\mathcal{K}_\varphi)$ solves (5.19) which again simplifies: Find $\dot{y}_\theta \in S^{\theta}_{y_\Omega}(\mathcal{K}_\varphi)$ such that for all $v \in S^{\theta}_{y_\Omega}(\mathcal{K}_\varphi)$ the following inequality holds true

$$\begin{aligned}\int_D \nabla \dot{y}_\theta^T \nabla(v - \dot{y}_\theta)\ \mathrm{d}x \geq -\int_D \nabla y_\Omega^T \left(\ \mathrm{div}(\theta)\mathrm{Id} - \nabla\theta - \nabla\theta^T\right) \nabla(v - \dot{y}_\theta) \\ +\ \mathrm{div}(\theta) f_\Omega(v - \dot{y}_\theta)\ \mathrm{d}x.\end{aligned} \tag{6.2}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Solving these two problems numerically requires more sophisticated techniques than the ones used in Chapter 4. We will have to operate on the matrices used to solve a given finite element problem. Thus to make notation easier we restrict ourselves to a FES of order 1 for all experiments in this chapter.

6.2. Solving the Obstacle Problem

We also had to solve problem (6.1) in Chapter 3. Previously we introduced an additional function $\lambda \in L^2(D)$ and solved a regularized version of this problem which looked like (3.10). This formulation had two advantages. Firstly we had to solve an equality instead of an inequality, and secondly we had to test for all $v \in H_0^1(D)$ instead of all $v \in \mathcal{K}_\varphi$. Both of these points make the application of standard algorithms in FEM much easier.

This time we do not want to apply any regularization and in order to solve (6.1) we make use of the techniques proposed in [6, Sections 5.2 and 5.6]. We first give a very brief summary of said techniques and then show how we can apply them to our specific problem. Note that in [6] the author handle the case of a lower obstacle set whereas we have an upper obstacle set. Therefore in the following there are slight changes compared to [6].

Given a vector $\ell \in (\mathbb{R} \cup \{+\infty\})^m$, a set

$$\Omega_B := \{x \in \mathbb{R}^m : x_i \leq \ell_i \text{ for all } i = 1, ..., m\},$$

a symmetric, positive definite matrix $A \in \mathbb{R}^{m \times m}$ and a vector $b \in \mathbb{R}^m$ we want to solve the minimization problem

$$\min_{x \in \Omega_B} \frac{1}{2} x^T A x - x^T b. \tag{6.3}$$

To do so the author in [6] propose the Gradient Projection Method, for which we need a few additional definitions. First we introduce the projection P_{Ω_B}, which is defined by

$$P_{\Omega_B} : \mathbb{R}^m \to \Omega_B : x \mapsto \min\{\ell, x\} \text{ component-wise}$$

and the gradient g by

$$g : \mathbb{R}^m \to \mathbb{R}^m : x \mapsto Ax - b.$$

We define the active set $\mathfrak{A}(x)$ and the free set $\mathfrak{F}(x)$ for $x \in \mathbb{R}^m$ by

$$\mathfrak{A}(x) := \{i \in \{1, ..., m\} : x_i = \ell_i\} \text{ and } \mathfrak{F}(x) := \{i \in \{1, ..., m\} : x_i \neq \ell_i\}.$$

Using these two sets we can further define two functions β, $\phi : \mathbb{R}^m \to \mathbb{R}^m$ componentwise by

$$\begin{aligned} \text{for } i \in \mathfrak{A}(x): &\quad \beta_i(x) = \max\{g_i(x), 0\} &&\text{and} \quad \phi_i(x) = 0, \\ \text{and for } i \in \mathfrak{F}(x): &\quad \beta_i(x) = 0 &&\text{and} \quad \phi_i(x) = g_i(x). \end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Finally we define the projected gradient $g^P(x) := \beta(x) + \phi(x)$.

Using all this we can state the algorithm to solve (6.3).

Algorithm 2 Gradient Projection Method

INPUT: $A \in \mathbb{R}^{m\times m}$ symmetric positive definite, $b \in \mathbb{R}^m$, $\ell \in (\mathbb{R}\cup\{+\infty\})^m$ and tol > 0

OUTPUT: Approximate solution $\widetilde{x} \in \mathbb{R}^m$ of (6.3)

$n \leftarrow 0$
Choose some arbitrary $x_0 \in \Omega_B$ and $\overline{\alpha} \in (0, 2\|A\|^{-1})$
while $\|g^P(x_k)\| \geq$ tol **do**
  $x_{k+1} \leftarrow P_{\Omega_B}(x_k - \overline{\alpha} g(x_k))$
  $k \leftarrow k+1$
end while

RETURN: $\widetilde{x} \leftarrow x_k$.

The key observation on solving (6.1) numerically now is that on a descrete level solving (6.1) is equivalent to solving (6.3) for the correct matrix $A \in \mathbb{R}^{m\times m}$ and vectors $b, \ell \in \mathbb{R}^m$ as can be seen with the same techniques as in Theorem 5.11. Thus we now shift the focus on the discretization of our problem. We explain our approach by solving (6.1).

Let $\mathcal{T}$ be a triangulation of our domain D consisting of triangles T and let $\mathcal{P}_1$ be the space of polynomials of order 1. Let

$$S^{1,1}(\mathcal{T}) := \{u \in H^1(D) : u|_K \in \mathcal{P}_1 \text{ for all } K \in \mathcal{T}\}$$

and $(\psi_i)_{i=1,\dots,m} \subseteq S_0^{1,1}(\mathcal{T}) := S^{1,1}(\mathcal{T}) \cap H_0^1(D)$ be a basis of hat functions of $S_0^{1,1}(\mathcal{T})$, i.e. they satisfy $\psi_i(\underline{\mathbf{x}}_j) = \delta_{ij}$ for all $i = 1, \dots, m$ and all vertices $\underline{\mathbf{x}}_j \in \mathbb{R}^d$, $j = 1, \dots, m$ of our triangulation $\mathcal{T}$.

Using this basis we assemble the stiffness matrix A by calculating

$$A_{i,j} := a(\psi_j, \psi_i) = \int_D \nabla\psi_j^T \nabla\psi_i \, \mathrm{d}x \qquad \text{for all } i, j = 1, \dots, m$$

and the load vector b by calculating

$$b_i := \int_D f_\Omega \psi_i \, \mathrm{d}x \qquad \text{for all } i = 1, \dots, m.$$

We also have to assemble our obstacle vector $\ell \in \mathbb{R}^m$ by projecting φ_Ω onto $S^{1,1}(\mathcal{T})$.

In order to apply Algorithm 2 we have to do one additional step. Since we want our solution to be in $S_0^{1,1}(\mathcal{T})$ we have to remove all rows and columns in the matrix A and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the vectors b and ℓ corresponding to vertices $\underline{\mathbf{x}} \in \partial D$. After applying the Gradient Projection Method we add the removed indices back into the solution vector $\widetilde{x}$ by filling it with zeros on the corresponding places. This way we ensure that our solution is indeed an element of $S_0^{1,1}(\mathcal{T})$ and not just $S^{1,1}(\mathcal{T})$.

To sum up the complete algorithm to solve (6.1) reads as follows.

Algorithm 3 Obstacle Problem for FEM of order 1

INPUT: Bounded, coercive symmetric bilinear form $a(.,.)$, $f_\Omega \in L^2(D)$, obstacle φ_Ω, hat basis $(\psi_i)_{i=1,...,m}$ of $S_0^{1,1}(\mathcal{T})$, tol > 0
OUTPUT: Approximate solution $\mathbf{y}_\Omega \in S^{1,1}(\mathcal{T})$ of (5.11).

Assemble $A \in \mathbb{R}^{m\times m}$ by calculating $A_{i,j} = a(\psi_j, \psi_i)$ for all $i, j = 1, ..., m$.
Assemble $b \in \mathbb{R}^m$ by calculating $b_i = \int_D f_\Omega \psi_i \, \mathrm{d}x$ for all $i = 1, ..., m$.
Assemble $\ell \in \mathbb{R}^m$ by calculating $\ell_i = \varphi_\Omega(\underline{\mathbf{x}}_i)$ for all $i = 1, ..., m$.
Determine the set $\mathfrak{B}$ containing all indices $i \in \{1, ..., m\}$ for which $\underline{\mathbf{x}}_i \in \partial D$.
Remove all rows and columns $i \in \mathfrak{B}$ from A, b and ℓ.
$\mathbf{y} \leftarrow$ GradientProjectionMethod$(A,\ b,\ \ell,\ \text{tol})$
$\mathbf{y}_\Omega \leftarrow 0 \in \mathbb{R}^m$
$\mathbf{y}_\Omega \leftarrow \widetilde{y}$ for all indices $i \in \{1, ..., m\} \backslash \mathfrak{B}$

RETURN: $\mathbf{y}_\Omega$.

In Figure 6.1 we compare the solution $y_{\gamma,c,\Omega}$ of the regularized obstacle problem of the target function with the unregularized solution y_Ω. For increasing γ and c we observe that $y_{\gamma,c,\Omega}$ converges to y_Ω in $H_0^1(D)$. Note that for higher γ and c the numerical errors add up too much and thus the Newton solver used to obtain $y_{\gamma,c,\Omega}$ does not reach a higher accuracy. We use a mesh with a maximum element diameter of 0.05, the obstacle was φ_2 and tol $= 10^{-11}$. The code to demonstrate the algorithm can be found in Appendix A.5.

6.3. Calculating the Material Derivative

Solving (6.2) proves to be harder then (6.1). The main difficulty is that we are looking for a solution $\dot{y}_\theta$ in the space $S^\theta_{y_\Omega}(\mathcal{K}_\varphi)$ which we somehow need to grasp numerically. To achieve this the key observation is the equivalence in (5.20) which we exploit.

The idea is the following: (6.2) looks quite similar to an obstacle problem such as (6.1) if it were not for the odd space $S^\theta_{y_\Omega}(\mathcal{K}_\varphi)$ in which we want to find a solution. Due to (5.20) we are able to equivalently express this problem as an obstacle problem over $S_0^{1,1}(\mathcal{T})$ by introducing a specific obstacle, which only depends on the active set. We then again solve this new obstacle problem by using the Gradient Projection Method.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

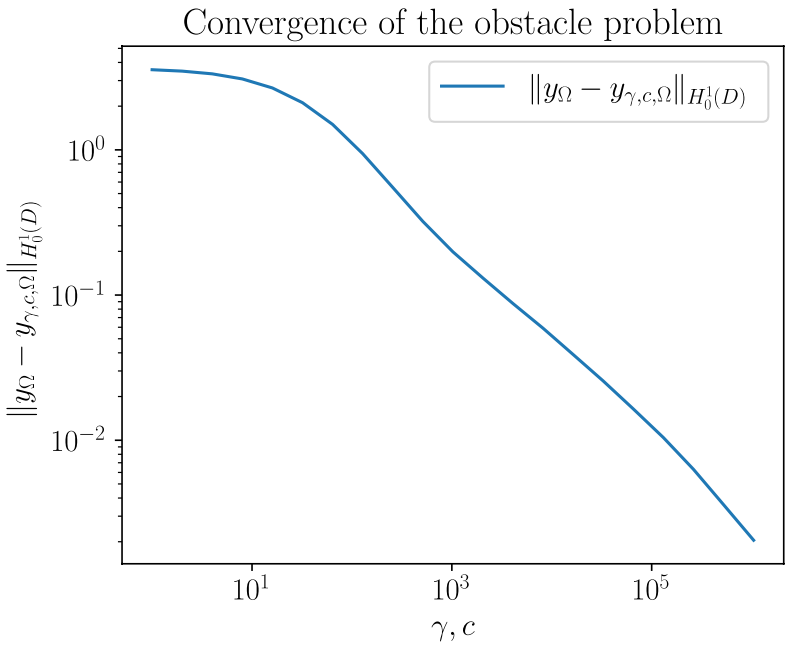


Figure 6.1.: $\|y_{\gamma,c,\Omega} - y_\Omega\|_{H_0^1(D)}$ for the target functions for various γ, c and obstacle φ_2. The shape Ω is depicted in Figure 4.1. We use a mesh with a maximum element diameter of 0.05 and tol $= 10^{-11}$ in Algorithm 3.

One necessary condition for our material derivatice $\dot{y}_\theta$ to be an element of $S^\theta_{y_\Omega}(\mathcal{K}_\varphi)$ is that it satisifies $\langle \mathbf{A}(y_\Omega), \dot{y}_\Omega \rangle_{H_0^1(D)} = 0$ according to (5.20), i.e. $\dot{y}_\theta \in \operatorname{kern}(\mathbf{A}(y_\Omega))$. Note that in our case the obstacle φ is independent of the shape and thus $\dot{\varphi} = 0$. This implies that we have to determine the kernel of $\mathbf{A}(y_\Omega)$. The kernel is a linear subspace of $S_0^{1,1}(\mathcal{T})$ and for each of our basis functions ψ_i, $i = 1, ..., m$ we want to find out if $\psi_i \in \operatorname{kern}(\mathbf{A}(y_\Omega))$. To do so we calculate $a(y_\Omega, \psi_i) + \int_D f_\Omega \psi_i \, \mathrm{d}x$ for every $i = 1, ..., m$. By definition of the kernel and our operator $\mathbf{A}(y_\Omega)$ if this expression is 0, then $\psi_i \in \operatorname{kern}(\mathbf{A}(y_\Omega))$. Due to numerical errors it is not exactly zero in our experiments but if it is close enough, i.e. smaller than some given tolerance tol, we put the index into a set $\mathfrak{K}$ containing all indices of basis functions which are an element of the kernel.

Next we focus on the other condition defining $S^\theta_{y_\Omega}(\mathcal{K}_\varphi)$ in (5.20). We need to calculate the active set, i.e. the set $\{x \in D : y_\Omega(x) = \varphi_\Omega(x)\}$. In the descrete setting we determine all indices $i \in \{1, ..., m\}$ for which $y_\Omega(\underline{\mathbf{x}}_i) = \varphi_\Omega(\underline{\mathbf{x}}_i)$. Using those indices and since we have a shape-independent obstacle we can define an obstacle

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

vector $\ell \in \mathbb{R}^m$ by

$$\ell_i := \begin{cases} 0 & \text{if } y_\Omega(\underline{\mathbf{x}}_i) = \varphi_\Omega(\underline{\mathbf{x}}_i), \\ +\infty & \text{otherwise,} \end{cases} \qquad \text{for all } i = 1, ..., m.$$

We also determine the set $\mathfrak{B}$ of boundary vertices again as we did in the previous section. We calculate the stiffness matrix A and the load vector b of (6.2) as before. Note though, that calculating b involves more terms now since it also includes the dependency on θ of the material derivative.

We now have everything we need to calculate the material derivative $\dot{y}_\theta$ numerically. We can write our numerical approximation $\dot{\mathbf{y}}_\theta \in S_0^{1,1}(\mathcal{T})$ of $\dot{y}_\theta$ as $\dot{\mathbf{y}}_\theta = \sum_{i=1}^m \dot{\mathbf{y}}_\theta^i \psi_i$. Since it must be an element of $\operatorname{kern}(\mathbf{A}(y_\Omega))$ all $\dot{\mathbf{y}}_\theta^i$ for $i \notin \mathfrak{K}$ must be 0. Thus we remove all rows and coumns $i \notin \mathfrak{K}$ of A, b and ℓ since they do not contribute to our solution. We also remove all rows and columns $i \in \mathfrak{B}$ which correspond to the boundary since our solution has to be an element of $S_0^{1,1}(\mathcal{T})$ and therefore they must be 0 as well. Now we can apply Algorithm 2. It gives us a solution which is an element of the kernel since we reduced A, b and ℓ to the kernel. It also satisifies $\dot{\mathbf{y}}_\theta \leq 0$ q.e. on $\{y_\Omega = \varphi_\Omega\}$ since that is our artificial obstacle and the result of the Gradient Projection Method gives us the solution satisfying this condition. As a last step we fill the resulting vector of the Gradient Projection Method with zeros at all removed indices and have a solution of (6.2).

In Algorithm 4 we sum up our approach.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithm 4 Material Derivative for FEM of Order 1

INPUT: Bounded, coercive symmetric bilinear form $a(.,.)$, $y_\Omega \in H_0^1(D)$ solution of (5.11) , $f_\Omega \in L^2(D)$, obstacle φ_Ω, hat basis $(\psi_i)_{i=1,...,m}$ of $S_0^{1,1}(\mathcal{T})$, tol > 0
OUTPUT: Approximate solution $\mathbf{y}_\theta \in S_0^{1,1}(\mathcal{T})$ of (5.11).

Assemble $A \in \mathbb{R}^{m\times m}$ by calculating $A_{i,j} = a(\psi_j, \psi_i)$ for all $i, j = 1, ..., m$.
Assemble $\underline{f} \in \mathbb{R}^m$ by calculating $\underline{f}_i = \int_D f_\Omega \psi_i \, \mathrm{d}x$ for all $i = 1, ..., m$.
Assemble $\underline{b} \in \mathbb{R}^m$ by calculating the right-hand side of (6.2) tested with $(\psi_i)_{i=1,...,m}$.
Assemble $\ell \in \mathbb{R}^m$ by calculating $\ell_i = \varphi_\Omega(\underline{\mathbf{x}}_i)$ for all $i = 1, ..., m$.
Assemble $\widetilde{y}_\Omega \in \mathbb{R}^m$ by calculating $(\widetilde{y}_\Omega)_i = y_\Omega(\underline{\mathbf{x}}_i)$ for all $i = 1, ..., m$.

$\mathfrak{K} \leftarrow \{\}$
for $i = 1, ..., m$ **do**
  if $|\mathbf{y}_\Omega^T A e_i - f_i| \leq$ tol **then**
    $\mathfrak{K} \ni i$
  end if
end for

$\ell \leftarrow 0 \in \mathbb{R}^m$
for $i = 1, ..., m$ **do**
  if $|(\mathbf{y}_\Omega)_i - \varphi_\Omega(\underline{\mathbf{x}}_i)| >$ tol **then**
    $\ell_i \leftarrow +\infty$
  end if
end for

Remove all rows and columns $i \notin \mathfrak{K}$ from A, b and ℓ.
Determine the set $\mathfrak{B}$ containing all indices $i \in \{1, ..., m\}$ for which $\underline{\mathbf{x}}_i \in \partial D$.
Remove all rows and columns $i \in \mathfrak{B}$ from A, b and ℓ.
$\mathbf{y} \leftarrow$ GradientProjectionMethod(A, b, ℓ, tol)
$\dot{\mathbf{y}}_\theta \leftarrow 0 \in \mathbb{R}^m$
$\dot{\mathbf{y}}_\theta \leftarrow \mathbf{y}$ for all indices $i \in \mathfrak{K}$

RETURN: $\widetilde{y}_\theta$.

In Figure 6.2 we see the results of a Taylor test similar to the one performed in Section 4.2.2. Again we set φ_2 as an obstacle and the target function is the solution of the obstacle problem on the target configuration calculated as outlined in this chapter. Ω is given by the start configuration, i.e. a circle of radius 0.3 centered around $M_1 = (0.5, 0.5)$. Figure 6.2 depicts the local error, therefore we expect the shape derivative to converge at least with order 2 again as is the case.

In Figure 6.3 we perform the same experiment as in Figure 6.2 but we calculate

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

the shape derivative (5.21) without the term containing the material derivative. No convergence is obtained in this case which shows that this term is not negligible in general and it indicates the soundness of Algorithm 4. The code for this section can be found in Appendix A.6.

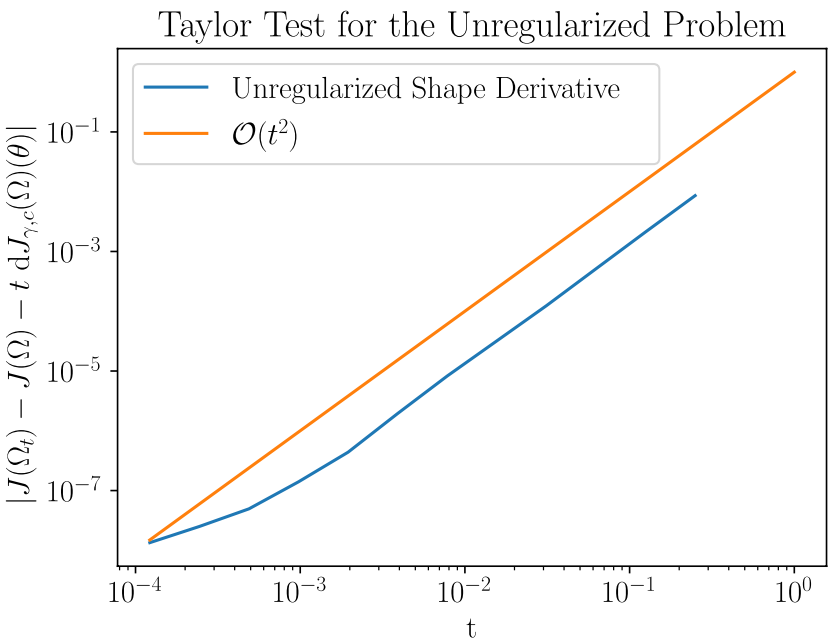


Figure 6.2.: Taylor test with obstacle φ_2 and a maximum element diameter of our mesh of 0.01. The shape Ω is given by the circle of radius 0.3 around $M_2 = (0.5, 0.5)$.

6.4. Solving the Shape Optimization Problem

Another difficulty arises when it comes to solving the shape optimization problem in the unregularized case as we can not apply Algorithm 1 directly. In Chapter 4 we used the fact that $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ was linearly dependent of θ. Thus obtaining the optimal decent direction θ was internally done by solving a system of linear equations. In contrast in the unregularized case we have a non-linear dependence of the material derivative $\dot{y}_\theta$ on θ which we need in order to calculate $\mathrm{d}J(\Omega)(\theta)$. We also operate on the matrix level of our FEM problem instead of some higher functionality provided by NGSolve.

The basis for our approach is Algorithm 1. To calculate the optimal decent direction we need $\mathrm{d}J(\Omega)(\theta)$, which consists of the two terms given in (5.21). The first

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

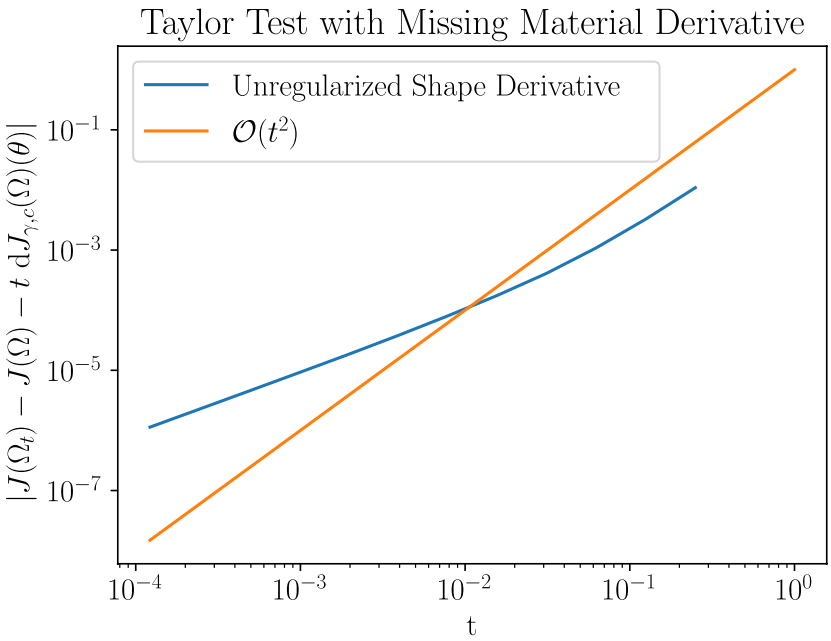


Figure 6.3.: Taylor test with obstacle φ_2 and a maximum element diameter of our mesh of 0.01. The shape Ω is given by the circle of radius 0.3 around $M_2 = (0.5, 0.5)$. Without the term containing the material derivative in (5.21) no convergence is achieved.

term poses no problem, as it is linearly dependent on θ. For the second term we do not try for all θ but instead use the vector field of the previous step. This way the second term is just a constant in every step. If the changes in the vector field are small enough then we still get a suitable decent direction for J. This is due to Theorem 6.1. Of course in general it will not be the best one. In our implementation in NGSolve we thus define in the n-th step the shape derivative

$$\mathrm{d}J(\Omega_n)(\theta) := \langle d_\Phi \mathcal{J}(Id, y_{\Omega_n}), \theta\rangle_{C^{0,1}(D;\mathbb{R}^d)} + \left\langle d_y \mathcal{J}(Id, y_{\Omega_n}), \dot{y}_{\theta_{n-1}}\right\rangle_{H^1(\mathrm{supp}(\theta))}, \tag{6.4}$$

where $\mathrm{supp}(\theta)$ denotes the support of θ. The adapted algorithm we propose is given in Algorithm 5.

Theorem 6.1. *Let $D \subset \mathbb{R}^d$ be bounded, $\Omega \subset \mathbb{R}^d$ and $(\theta_n)_{n\in\mathbb{N}}$ and θ in $C_c^1(D, \mathbb{R}^d)$ such that $\theta_n \to \theta$ as $n \to +\infty$. Let $\dot{y}_{\theta_n}$ and $\dot{y}_\theta$ be defined by (6.2).*

Then $\dot{y}_{\theta_n} \to \dot{y}_\theta$ in $H_0^1(D)$ as $n \to +\infty$.

Proof. Since both $\dot{y}_{\theta_n}$ and $\dot{y}_\theta$ are solutions of (6.2) and $S_{y_\Omega}^{\theta_n}(\mathcal{K}_\varphi) = S_{y_\Omega}^{\theta}(\mathcal{K}_\varphi)$ in our case of a shape-independent obstacle which can be obtained immediately from (5.20),

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

we can test both functions with $\dot{y}_\theta$ and $\dot{y}_{\theta_n}$ respectively to obtain

$$\begin{aligned}\int_D \nabla \dot{y}_{\theta_n}^T \nabla(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x \geq -\int_D \nabla y_\Omega^T \left(\operatorname{div}(\theta_n)\mathrm{Id} - \nabla\theta_n - \nabla\theta_n^T\right)\nabla(\dot{y}_\theta - \dot{y}_{\theta_n}) \\ +\ \operatorname{div}(\theta_n) f_\Omega(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x\end{aligned} \tag{6.5}$$

$$\begin{aligned}\int_D \nabla \dot{y}_\theta^T \nabla(\dot{y}_{\theta_n} - \dot{y}_\theta)\,\mathrm{d}x \geq -\int_D \nabla y_\Omega^T \left(\operatorname{div}(\theta)\mathrm{Id} - \nabla\theta - \nabla\theta^T\right)\nabla(\dot{y}_{\theta_n} - \dot{y}_\theta) \\ +\ \operatorname{div}(\theta) f_\Omega(\dot{y}_{\theta_n} - \dot{y}_\theta)\,\mathrm{d}x.\end{aligned} \tag{6.6}$$

We multiply the second inequality by -1

$$\begin{aligned}\int_D \nabla \dot{y}_\theta^T \nabla(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x \leq -\int_D \nabla y_\Omega^T \left(\operatorname{div}(\theta)\mathrm{Id} - \nabla\theta - \nabla\theta^T\right)\nabla(\dot{y}_\theta - \dot{y}_{\theta_n}) \\ +\ \operatorname{div}(\theta) f_\Omega(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x,\end{aligned}$$

subtract (6.5) from (6.6) and get

$$\begin{aligned}&\int_D \nabla(\dot{y}_\theta - \dot{y}_{\theta_n})^T \nabla(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x \\ &\quad\leq -\int_D \nabla y_\Omega^T \left(\operatorname{div}(\theta - \theta_n)\mathrm{Id} - \nabla(\theta - \theta_n) - \nabla(\theta - \theta_n)^T\right)\nabla(\dot{y}_\theta - \dot{y}_{\theta_n}) \\ &\qquad +\ \operatorname{div}(\theta - \theta_n) f_\Omega(\dot{y}_\theta - \dot{y}_{\theta_n})\,\mathrm{d}x.\end{aligned}$$

Using the Poincaré inequality with constant $C > 0$ and Hölder's inequality we can further calculate

$$\begin{aligned}\|\nabla(\dot{y}_\theta - \dot{y}_{\theta_n})\|_{L^2(D)} \leq C\|\operatorname{div}(\theta - \theta_n)\mathrm{Id} - \nabla(\theta - \theta_n) - \nabla(\theta - \theta_n)^T\|_\infty \|y_\Omega\|_{L^2(D)} \\ + C\|\operatorname{div}(\theta - \theta_n)\|_\infty \|f_\Omega\|_{L^2(D)}.\end{aligned}$$

The right-hand side converges to 0 as $n \to \infty$ by assumption and again due to Poincaré we have $\frac{1}{C}\|\nabla(\dot{y}_\theta - \dot{y}_{\theta_n})\|_{L^2(D)} \leq \|\dot{y}_\theta - \dot{y}_{\theta_n}\|_{L^2(D)}$ and thus $\dot{y}_{\theta_n} \to \dot{y}_\theta$ in $H_0^1(D)$. $\square$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithm 5 Gradient Algorithm for the Unregularized Problem

INPUT: Initial domain Ω_0, target function $\overline{y}$, obstacle φ, constants $\alpha, \text{tol} > 0$, $N_{max} > 0$, $\zeta \geq 0$.
OUTPUT: Solution of (3.3).

$n \leftarrow 0$
$\theta_0 \leftarrow 0$
$\dot{y}_{\theta_0} \leftarrow 0$
do
  Calculate y_{Ω_n} by solving (6.1).
  Calculate $\dot{y}_{\theta_n}$ by solving (6.2).
  Assemble $\mathrm{d}J(\Omega_n)(.)$ as given in (6.4).
  Calculate $\nabla^H g(\theta_n)$ by solving (4.9).
  while TRUE **do**
    if $J((\mathrm{Id} - \alpha\nabla^H g(\theta_n)(\Omega_n)) < J(\Omega_n) - \zeta\alpha\|\nabla^H g(\theta)\|^2$ **then**
      $\Omega_{n+1} \leftarrow (\mathrm{Id} - \alpha\nabla^H g(\theta_n))(\Omega_n)$
      $\theta_{n+1} \leftarrow \theta_n - \alpha\nabla^H g(\theta_n)$
      $n \leftarrow n + 1$
      Increase α.
      Break inner while-loop.
    else
      Decrease α.
    end if
  end while
while $n < N_{max}$ and $\|\nabla g(\Omega_n)\| > \text{tol}$

RETURN: Ω_n.

In Figure 6.4 we see the results of our shape optimization problem for the various Hilbert spaces given in (4.10). We used a mesh with a maximum element diameter of 0.02. The target function was calculated using Algorithm 3. The constants are given by $\alpha = 0.5$, $\text{tol} = 10^{-11}$, $N_{max} = 100$ and $\zeta = 10^{-4}$. In the case of H_3 we set $\gamma_{CR} = 10$. Again we notice that in the case of H_1 and H_2 the algorithm does not terminate. The code for this section can be found in Appendix A.7.

At this point we have to draw attention to a few caveats regarding Algorithm 5. As we have seen in Figure 6.3 in general the term containing the material derivative in (5.21) is not negligible in order to calculate the shape derivative properly. In all our shape optimization experiments however this term had little to no impact on the results and we achieved convergence even without it. This could be due to a multitude of reasons ranging from an inadequate problem definition to using a mesh that is too coarse or some fundamental flaw in our proposed Algorithm 5. Further

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

research needs to be conducted to obtain a deeper understanding of this behaviour.

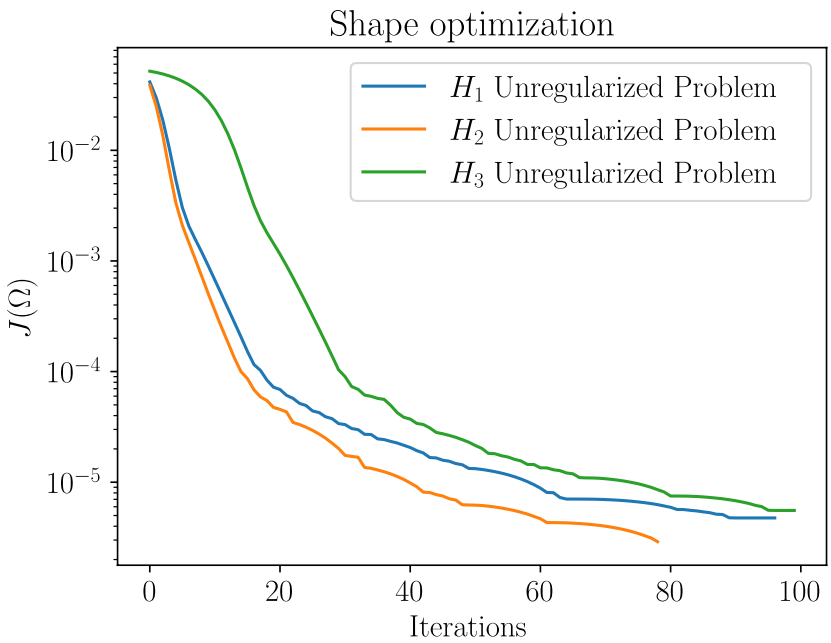


Figure 6.4.: Convergence of the shape optimization problem for various Hilbert spaces. Using H_1 and H_2 Algorithm 5 has to be stopped before it finishes 100 iterations as it runs into a loop.

6.5. Comparing the Regularized with the Unregularized Problem

In this section we want to compare the various approaches we discussed in this thesis to solve the shape optimization problem numerically. The code can be found in Appendix A.8.

We begin by introducing yet another approach to solve (4.2). The shape derivative formula (5.21) is not restricted to the unregularized case and can also be applied to the regularized case. To do so we first derive a formula for the material derivative of (4.5) and then prove a stronger version of Theorem 6.1 to ensure that the application of Algorithm 5 is justified.

Theorem 6.2. *Let $y_{\gamma,c,\Omega}$ be the solution of (4.5) and $\theta \in C_c^1(D, \mathbb{R}^d)$. Let $M(t)$ be given by (5.13) with $M = \mathrm{Id}$ and let $\overline{\lambda} \in H^1(D)$. Then the material derivative $\dot{y}_{\gamma,c,\theta}$*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

of $y_{\gamma,c,\Omega}$ is given by the solution of: Find $\dot{y}_{\gamma,c,\theta} \in H_0^1(D)$ such that for all $v \in H_0^1(D)$

$$\begin{aligned}\int_D \nabla \dot{y}_{\gamma,c,\theta}^T \nabla v \,\mathrm{d}x + \int_D c\, \max_\gamma'(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))\dot{y}_{\gamma,c,\theta} v \,\mathrm{d}x \\ = -\int_D M'(0)\nabla y_{\gamma,c,\Omega}^T \nabla v \,\mathrm{d}x - \int_D \mathrm{div}(\theta)\max_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))v \,\mathrm{d}x \\ - \int_D \max_\gamma'(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\nabla\overline{\lambda}\cdot\theta)v \,\mathrm{d}x \\ + \int_D \dot{f}_\theta v \,\mathrm{d}x + \int_D \mathrm{div}(\theta) f_\Omega v \,\mathrm{d}x\end{aligned} \tag{6.7}$$

holds true.

Proof. The solution $x_{\gamma,c,\Omega_t} \in H_0^1(D)$ of (4.5) on the perturbed domain Ω_t solves

$$\int_D \nabla y_{\gamma,c,\Omega_t}^T \nabla v \,\mathrm{d}x + \int_D \max_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega_t} - \varphi))v \,\mathrm{d}x = \int_D f_{\Omega_t} v \,\mathrm{d}x$$

for all $v \in H_0^1(D)$. Transforming this equation back to the unperturbed problem this is equivalent to solving

$$\begin{aligned}\int_D M(t)\nabla(y_{\gamma,c,\Omega_t} \circ \Phi_t^\theta)^T \nabla v \,\mathrm{d}x + \int_D \xi(t)\max_\gamma(\overline{\lambda} \circ \Phi_t^\theta + c((y_{\gamma,c,\Omega_t} \circ \Phi_t^\theta) - \varphi))v \,\mathrm{d}x \\ = \int_D \xi(t)(f_{\Omega_t} \circ \Phi_t^\theta)v \,\mathrm{d}x\end{aligned}$$

for all $v \in H_0^1(D)$. Deriving this equation with respect to t, using the product rule, the chain rule, the definition of the material derivative, the theorems from Section 2.3 and evaluating the result at $t = 0$ we get

$$\begin{aligned}&\int_D \nabla \dot{y}_{\gamma,c,\theta}^T \nabla v \,\mathrm{d}x + \int_D M'(0)\nabla y_{\gamma,c,\Omega}^T \nabla v \,\mathrm{d}x \\ &+ \int_D \mathrm{div}(\theta)\max_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))v \,\mathrm{d}x \\ &+ \int_D \max_\gamma'(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\nabla\overline{\lambda}\cdot\theta + c\,\dot{y}_{\gamma,c,\theta})v \,\mathrm{d}x \\ &\qquad = \int_D \mathrm{div}(\theta) f_\Omega v \,\mathrm{d}x + \int_D \dot{f}_\theta v \,\mathrm{d}x.\end{aligned}$$

Rearranging the individual terms finally yields the result. □

Theorem 6.3. *Let $D \subset \mathbb{R}^d$ be bounded, $c > 0$, $(\theta_n)_{n\in\mathbb{N}}$ and $\theta \in C_c^1(D, \mathbb{R}^d)$ such that $\theta_n \to \theta$ in $C_c^1(D, \mathbb{R}^d)$ as $n \to +\infty$. Let $\dot{y}_{\gamma,c,\theta_n}$ be the solutions of (6.7) for each $n \in \mathbb{N}$. Let $\dot{f}_{\theta_n} \to \dot{f}_\theta$ in $L^2(D)$ as $n \to \infty$.*

Then $\dot{y}_{\gamma,c,\theta_n} \to \dot{y}_{\gamma,c,\theta}$ in $H_0^1(D)$ as $n \to +\infty$ where $\dot{y}_{\gamma,c,\theta}$ is the solution of (6.7).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Proof. We test (6.7) with $\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}$ for both $\dot{y}_{\gamma,c,\theta_n}$ and $\dot{y}_{\gamma,c,\theta}$ and subtract both equations from eachother. This yields

$$\begin{aligned}
&\int_D \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})^T \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n})\,\mathrm{d}x \\
&\quad = -\int_D c\,\max'_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n})\,\mathrm{d}x \\
&\qquad - \int_D (\mathrm{div}(\theta_n - \theta)\mathrm{Id} - \nabla(\theta_n - \theta)^T - \nabla(\theta_n - \theta))^{-T} \nabla y^T_{\gamma,c,\Omega} \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad - \int_D \mathrm{div}(\theta_n - \theta)\mathrm{max}_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad - \int_D \mathrm{max}'_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\nabla\overline{\lambda} \cdot (\theta_n - \theta))(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad + \int_D (\dot{f}_{\theta_n} - \dot{f}_\theta)(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n})\,\mathrm{d}x + \int_D \mathrm{div}(\theta_n - \theta) f_\Omega(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x.
\end{aligned}$$

Due to Assumption 3.2 (iii) and $c > 0$ we know that the first term containing max'_γ is greater or equal to 0 almost everywhere and therefore

$$\begin{aligned}
&\int_D \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})^T \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n})\,\mathrm{d}x \\
&\quad \le -\int_D (\mathrm{div}(\theta_n - \theta)\mathrm{Id} - \nabla(\theta_n - \theta)^T - \nabla(\theta_n - \theta)^{-T}) \nabla y^T_{\gamma,c,\Omega} \nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad - \int_D \mathrm{div}(\theta_n - \theta)\mathrm{max}_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad - \int_D \mathrm{max}'_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))(\nabla\overline{\lambda} \cdot (\theta_n - \theta))(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x \\
&\qquad + \int_D (\dot{f}_{\theta_n} - \dot{f}_\theta)(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n})\,\mathrm{d}x + \int_D \mathrm{div}(\theta_n - \theta) f_\Omega(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\,\mathrm{d}x.
\end{aligned}$$

We can further estimate

$$\begin{aligned}
&\|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|^2_{L^2(D)} \\
&\quad \le \|\mathrm{div}(\theta_n - \theta)\mathrm{Id} - \nabla(\theta_n - \theta)^T - \nabla(\theta_n - \theta)^{-T}\|_\infty \|\nabla y_{\gamma,c,\Omega}\|_{L^2(D)} \\
&\qquad\qquad \|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|_{L^2(D)} \\
&\qquad + \|\mathrm{div}(\theta_n - \theta)\|_\infty \|\mathrm{max}_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))\|_{L^2(D)} \|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|_{L^2(D)} \\
&\qquad + \|\mathrm{max}'_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))\|_\infty \|\nabla\overline{\lambda} \cdot (\theta_n - \theta)\|_{L^2(D)} \|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|_{L^2(D)} \\
&\qquad + \|\dot{f}_{\theta_n} - \dot{f}_\theta\|_{L^2(D)} \|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta_n}\|_{L^2(D)} + \|\mathrm{div}(\theta_n - \theta)\|_\infty \|f_\Omega\|_{L^2(D)} \\
&\qquad\qquad \|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|_{L^2(D)}.
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Using the Poincaré inequality with constant $C > 0$, $\max'_\gamma \in [0,1]$ due to Assumption 3.2 (iii) and dividing both sides by $\|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|_{L^2(D)}$ we get

$$\begin{aligned}
&\|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|_{L^2(D)} \\
&\qquad \leq \|\mathrm{div}(\theta_n - \theta)\mathrm{Id} - \nabla(\theta_n - \theta)^T - \nabla(\theta_n - \theta)^{-T}\|_\infty \|\nabla y_{\gamma,c,\Omega}\|_{L^2(D)} \\
&\qquad\quad + C\|\mathrm{div}(\theta_n - \theta)\|_\infty \|\max_\gamma(\overline{\lambda} + c(y_{\gamma,c,\Omega} - \varphi))\|_{L^2(D)} \\
&\qquad\quad + \|\nabla\overline{\lambda}\|_{L^2(D)}\|\theta_n - \theta\|_\infty \\
&\qquad\quad + C\|\dot{f}_{\theta_n} - \dot{f}_\theta\|_{L^2(D)} + C\|\mathrm{div}(\theta_n - \theta)\|_\infty \|f_\Omega\|_{L^2(D)}.
\end{aligned}$$

The right-hand side converges to 0 as $n \to \infty$ due to our assumptions. Therefore we have $\|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|_{L^2(D)} \to 0$ as $n \to \infty$ and due to Poincaré inequality we also have $\frac{1}{C}\|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|_{L^2(D)} \leq \|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|_{L^2(D)}$. Thus

$$\|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|^2_{H^1_0(D)} = \|\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta}\|^2_{L^2(D)} + \|\nabla(\dot{y}_{\gamma,c,\theta_n} - \dot{y}_{\gamma,c,\theta})\|^2_{L^2(D)} \to 0.$$

□

Due to this convergence result our modified shape derivative formula (6.4) also converges for the regularized case. In Figure 6.5 we compare the various shape optimization approaches. All experiments were done in a FES of order 1 with a maximum element diameter of 0.02 and solve the problem outlined in Section 4.1 with obstacle φ_2. In the regularized case we set $c, \gamma = 10000$. For the Hilbert space H_3 we set $\gamma_{CR} = 10$. The integral formula (4.6) performs best at the beginning in each Hilbert space but is then overtaken by the other approaches. Using (5.21) with the material derivative (6.7) of the regualarized problem yields the best results in our example. It is closely followed by the unregularized approach.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

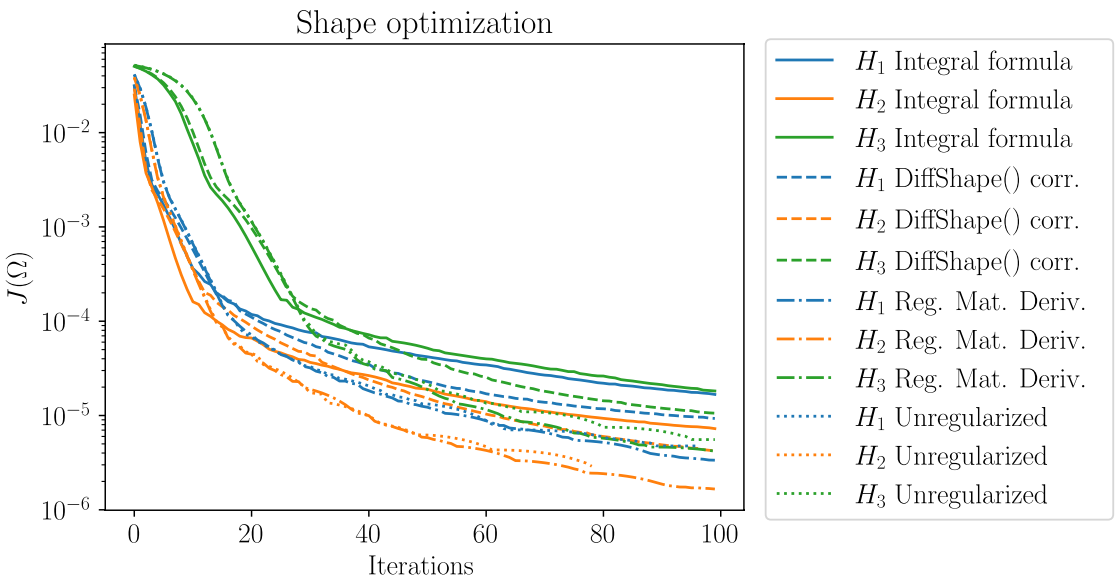


Figure 6.5.: Convergence of the shape optimization problem from Section 3.1 for various Hilbert spaces and various shape optimization approaches. For the regularized problems we set $\gamma, c = 10000$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7. Final Remarks and Outlook

We have reviewed two approaches on solving a shape optimization problem subject to a variational inequality. We have won new insights in the theory of a regularized approach and derived a new algorithm for solving the unregularized problem. We have seen, that both approaches have their advantages and disadvantages. First regularizing the variational inequality makes solving numerical problems much easier as we can use standard techniques in FEM. It is not clear however how the shape derivative of the regularized problem relates to the unregularized one. Some of these open questions are:

- What does z in Theorem 3.8 for an explicit $\max_\gamma$ look like? Does the regularized adjoint $p_{c,\Omega}$ converge to some p_Ω as $c \to +\infty$? If so, in which space does it converge? How does p_Ω relate to the unregularized problem?
- Is there a limiting object for $\mathrm{d}J_{\gamma,c}(\Omega)(\theta)$ as $c, \gamma \to +\infty$? How does it relate to $\mathrm{d}J(\Omega)(\theta)$ in the unregularized case?

The unregularized problem on the other side poses more questions in the aspect of a numerical implementation. While the theory can even be extended to a non-linear right-hand side in the variational inequality as was the case in [11], there is no algorithm to solve this specific problem yet. We have also had to limit ourselves to easier questions and thus some very naturally arising questions are:

- Can Algorithm 5 be extended both to handle non-static obstacles and higher order FEM?
- Can the algorithm be extended to solve non-linear problems which are investigated in [11]? What adaptions would need to be made in that case?

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

A. Code of Numerical Experiments

Here the code used in the numerical experiments can be found. Some functions can be found in the Auxiliary Functions.

A.1. Auxiliary Functions

```
from ngsolve import *
from netgen.geom2d import SplineGeometry
import numpy as np
import scipy as sp


def SimpleNewtonSolve(gfu, a, tol=1e-13, maxits=100):
    res = gfu.vec.CreateVector()
    du = gfu.vec.CreateVector()
    tmpfes = gfu.space
    for it in range(maxits):
        print("Iteration {:3}  ".format(it), end="")
        a.Apply(gfu.vec, res)
        a.AssembleLinearization(gfu.vec)
        du.data = a.mat.Inverse(tmpfes.FreeDofs()) * res
        gfu.vec.data -= du

        # stopping criteria
        stopcritval = sqrt(abs(InnerProduct(du, res)))
        print("<A u", it, ", A u", it, ">_{-1}^0.5 = ", stopcritval)
        if stopcritval < tol:
            return it


def max_gam(u, phi, c, gam, lam_bar):
    return IfPos(Norm(lam_bar + c * (u - phi)) - 1 / gam,
                 IfPos(lam_bar + c * (u - phi),
                       lam_bar + c * (u - phi),
                       0),
                 gam / 4 * (lam_bar + c * (u - phi)) * (lam_bar + c * (u - phi)) + 1 / 2 * (
                         lam_bar + c * (u - phi)) + 1 / (4 * gam))


def sign_gam(gfu, phi, c, gam, lam_bar):
    return IfPos(Norm(lam_bar + c * (gfu - phi)) - 1 / gam,
                 IfPos(lam_bar + c * (gfu - phi),
                       1,
                       0),
                 gam / 2 * (lam_bar + c * (gfu - phi)) + 1 / 2)


def getTargetMesh():
    geo = SplineGeometry()
    pnts = [(0.5, 0.2),
            (0.8, 0.2),
            (0.8, 0.5),
            (0.8, 0.8),
            (0.5, 0.8),
            (0.4, 0.8),
            (0.4, 0.7),
            (0.4, 0.6),
            (0.5, 0.6),
            (0.6, 0.6),
            (0.6, 0.5),
            (0.6, 0.4),
            (0.5, 0.4),
            (0.4, 0.4),
            (0.4, 0.3),
            (0.4, 0.2)]
    p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16 = [geo.AppendPoint(*pnt) for pnt in pnts]
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
    curves = [[["spline3", p1, p2, p3], "c1"],
              [["spline3", p3, p4, p5], "c1"],
              [["spline3", p5, p6, p7], "c1"],
              [["spline3", p7, p8, p9], "c1"],
              [["spline3", p9, p10, p11], "c1"],
              [["spline3", p11, p12, p13], "c1"],
              [["spline3", p13, p14, p15], "c1"],
              [["spline3", p15, p16, p1], "c1"]]

    for c, bc in curves:
        geo.Append(c, bc=bc, leftdomain=2, rightdomain=1)

    geo.AddRectangle(p1=(0, 0),
                     p2=(1, 1),
                     bc="rectangle",
                     leftdomain=1,
                     rightdomain=0)

    geo.SetMaterial(2, "w_int")
    geo.SetMaterial(1, "w_out")

    return geo


def getStartMesh(maxH):
    geo = SplineGeometry()
    geo.AddRectangle(p1=(0, 0),
                     p2=(1, 1),
                     bc="rectangle",
                     leftdomain=1,
                     rightdomain=0)
    geo.AddCircle(c=(0.5, 0.5),
                  r=0.3,
                  bc="g_int",
                  leftdomain=2,
                  rightdomain=1)
    geo.SetMaterial(1, "w_out")
    geo.SetMaterial(2, "w_int")
    geo.SetDomainMaxH(2, maxH)
    geo.SetDomainMaxH(1, maxH)

    return geo


def divVF(V):
    return V[0].Diff(x) + V[1].Diff(y)


def gradVF(V):
    gradV = CoefficientFunction((V[0].Diff(x), V[1].Diff(x), V[0].Diff(y), V[1].Diff(y)), dims=(2, 2))
    return gradV


def gradCF(phi):
    gradPhi = CoefficientFunction((phi.Diff(x), phi.Diff(y)), dims=(2, 1))
    return gradPhi


def IntegrandPaper(gfu, phi, c, gam, lam_bar, gfuTarget, gfp, f, V):
    return - (gfu - gfuTarget) * InnerProduct(grad(gfuTarget), V) \
           - InnerProduct(grad(gfu), (gradVF(V).trans + gradVF(V)) * grad(gfp)) \
           - c * sign_gam(gfu, phi, c, gam, lam_bar) * InnerProduct(gradCF(phi), V) * gfp \
           + sign_gam(gfu, phi, c, gam, lam_bar) * InnerProduct(gradCF(lam_bar), V) * gfp \
           - InnerProduct(gradCF(f), V) * gfp \
           + divVF(V) * (1 / 2 * (gfu - gfuTarget) ** 2
                         + InnerProduct(grad(gfu), grad(gfp))
                         + max_gam(gfu, phi, c, gam, lam_bar) * gfp - f * gfp)


def GetTargetFunction(maxH, phi):
    meshTarget = Mesh(getTargetMesh().GenerateMesh(maxh=maxH))
    meshTarget.Curve(4)

    fesTarget = H1(meshTarget, order=1, dirichlet="rectangle")
    uTarget = fesTarget.TrialFunction()
    wTarget = fesTarget.TestFunction()

    f_values = {"w_int": 100, "w_out": -10}
    fTarget = CoefficientFunction([f_values[mat] for mat in meshTarget.GetMaterials()])
    gffTarget = GridFunction(fesTarget)
    gffTarget.Set(fTarget)

    gfphiTarget = GridFunction(fesTarget)
    gfphiTarget.Set(phi)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
    # Fully regularized problem
    c = 3000000
    gam = 100000

    lam_barTarget = IfPos(fTarget, fTarget, 0)  # corresponds to lam_bar = max(0, f+\Delta \phi)
    gflam_barTarget = GridFunction(fesTarget)
    gflam_barTarget.Set(lam_barTarget)

    # Nonlinear variational formulation
    aTarget = BilinearForm(fesTarget, symmetric=False)
    aTarget += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
    aTarget += -fTarget * wTarget * dx(bonus_intorder=2)
    aTarget += max_gam(uTarget, phi, c, gam, lam_barTarget) * wTarget * dx(bonus_intorder=2)

    gfuTarget = GridFunction(fesTarget)
    SimpleNewtonSolve(gfuTarget, aTarget, maxits=100)
    return gfuTarget



def getObstacleProblemSolution(a, b, gfphi, mesh, fes, kmax=350000, tol=1e-11):
    coordinates = np.zeros((len(list(mesh.ngmesh.Points())), 3))
    for n, k in enumerate(list(mesh.ngmesh.Points())):
        coordinates[n, 0] = k[0]
        coordinates[n, 1] = k[1]
        coordinates[n, 2] = k[2]

    rows, cols, vals = a.mat.COO()
    A = sp.sparse.csr_matrix((vals, (rows, cols)))
    F = b.vec.FV().NumPy()[:]

    vecPhi = gfphi.vec.FV().NumPy()[:]

    # Indices of boundary coordinates
    boundaryCoordinates = []
    for k in range(len(coordinates)):
        if (coordinates[k, 0] == 0 or coordinates[k, 0] == 1 or coordinates[k, 1] == 0 or coordinates[k, 1] == 1):
            boundaryCoordinates.append(k)

    n = A.get_shape()[1]

    allCols = np.arange(A.shape[1])
    colsToKeep = np.where(np.logical_not(np.in1d(allCols, boundaryCoordinates)))[0]
    # print(colsToKeep)
    rows_to_keep = np.where(np.logical_not(np.in1d(allCols, boundaryCoordinates)))[0]
    A = A[:, colsToKeep]
    A = A[rows_to_keep, :]

    F = F[rows_to_keep]

    l = np.zeros((len(colsToKeep)))

    for i in range(len(colsToKeep)):
        l[i] = vecPhi[colsToKeep[i]]

    x0 = np.random.rand(len(colsToKeep))
    xold = x0
    xreturn = np.zeros((len(colsToKeep), 1))
    booleanA = abs(xold - l) > tol ** 2
    booleanF = abs(xold - l) < tol ** 2

    g = sp.sparse.csr_matrix.dot(A, xold) - F
    obs = np.multiply(g, booleanA)
    beta = np.multiply(np.fmin(g, 0), booleanF)

    gproj = obs + beta
    k = 0
    alphabar = 1 / 16
    while np.linalg.norm(gproj) > tol and k < kmax:
        xnew = np.minimum(xold - alphabar * (sp.sparse.csr_matrix.dot(A, xold) - F), l)
        k += 1
        xold = xnew
        xreturn = xold
        booleanA = abs(xold - l) > tol ** 2
        booleanF = abs(xold - l) < tol ** 2
        g = sp.sparse.csr_matrix.dot(A, xold) - F
        obs = np.multiply(g, booleanA)
        beta = np.multiply(np.maximum(g, 0), booleanF)
        gproj = obs + beta
        if (k % 1000 == 0):
            print(k)


    print(k)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
    result = np.zeros(n)
    for k in range(len(xreturn)):
        result[colsToKeep[k]] = xreturn[k]

    gfresult = GridFunction(fes)
    gfresult.vec.FV().NumPy()[:] = result

    return gfresult


def getMaterialDerivativeProblemSolution(a, b, gfphi, gfstate, indices, mesh, fes, kmax=350000, tol=1e-11):
    coordinates = np.zeros((len(list(mesh.ngmesh.Points())), 3))
    for n, k in enumerate(list(mesh.ngmesh.Points())):
        coordinates[n, 0] = k[0]
        coordinates[n, 1] = k[1]
        coordinates[n, 2] = k[2]

    rows, cols, vals = a.mat.COO()
    A = sp.sparse.csr_matrix((vals, (rows, cols)))
    F = b.vec.FV().NumPy()[:]
    n = A.get_shape()[1]

    vecPhi = gfphi.vec.FV().NumPy()[:]
    vecState = gfstate.vec.FV().NumPy()[:]
    vecObs = 10000*np.ones(n)
    for i in range(len(vecState)):
        if abs(vecPhi[i] - vecState[i]) < tol/2:
            vecObs[i] = 0

    # Indices of boundary coordinates
    boundaryCoordinates = []
    for k in range(len(coordinates)):
        if (coordinates[k, 0] == 0 or coordinates[k, 0] == 1 or coordinates[k, 1] == 0 or coordinates[k, 1] == 1):
            boundaryCoordinates.append(k)

    allCols = np.arange(A.shape[1])
    colsToKeepObs = np.where(np.logical_not(np.in1d(allCols, boundaryCoordinates)))[0]
    colsToKeep = np.intersect1d(colsToKeepObs, indices)
    A = A[:, colsToKeep]
    A = A[colsToKeep, :]

    F = F[colsToKeep]
    l = np.zeros((len(colsToKeep)))

    for i in range(len(colsToKeep)):
        l[i] = vecObs[colsToKeep[i]]

    x0 = -5*np.random.rand(len(colsToKeep))
    xold = x0
    xreturn = np.zeros((len(colsToKeep), 1))
    booleanA = abs(xold - l) > tol ** 2
    booleanF = abs(xold - l) < tol ** 2

    g = sp.sparse.csr_matrix.dot(A, xold) - F
    obs = np.multiply(g, booleanA)
    beta = np.multiply(np.fmin(g, 0), booleanF)

    gproj = obs + beta
    k = 0
    alphabar = 1 / 16
    while np.linalg.norm(gproj) > tol and k < kmax:
        xnew = np.minimum(xold - alphabar * (sp.sparse.csr_matrix.dot(A, xold) - F), l)
        k += 1
        xold = xnew
        xreturn = xold
        booleanA = abs(xold - l) > tol ** 2
        booleanF = abs(xold - l) < tol ** 2
        g = sp.sparse.csr_matrix.dot(A, xold) - F
        obs = np.multiply(g, booleanA)
        beta = np.multiply(np.maximum(g, 0), booleanF)
        gproj = obs + beta
        if(k % 1000 == 0):
            print(k)

    print(k)


    result = np.zeros(n)
    for j in range(len(xreturn)):
        result[colsToKeep[j]] = xreturn[j]

    gfresult = GridFunction(fes)
    gfresult.vec.FV().NumPy()[:] = result

    return gfresult
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

A.2. Target function

```
#################################################################################################################
#                                                                                                               #
# The aim of this program is to reproduce the Ω as found in the paper                                          #
# 1st test: Reproduce target shape from paper                                                                  #
# 2nd test: Solve fully regularized state equation on Ω for various λ̄                                          #
#                                                                                                               #
#################################################################################################################

from auxiliaryFunctions import SimpleNewtonSolve
from ngsolve import *
from netgen.geom2d import *
import numpy as np
import matplotlib.pyplot as plt

#################################################################################################################
#                                                FIRST TEST                                                     #
#################################################################################################################

# Create Γ using splines
geo = SplineGeometry()
pnts = [(0.5, 0.2), (0.8, 0.2), (0.8, 0.5), (0.8, 0.8), (0.5, 0.8), (0.4, 0.8), (0.4, 0.7), (0.4, 0.6), (0.5, 0.6),
        (0.6, 0.6), (0.6, 0.5), (0.6, 0.4), (0.5, 0.4), (0.4, 0.4), (0.4, 0.3), (0.4, 0.2)]
p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16 = [geo.AppendPoint(*pnt) for pnt in pnts]

# Create actual splines and name them
curves = [[["spline3", p1, p2, p3], "c1"],
          [["spline3", p3, p4, p5], "c1"],
          [["spline3", p5, p6, p7], "c1"],
          [["spline3", p7, p8, p9], "c1"],
          [["spline3", p9, p10, p11], "c1"],
          [["spline3", p11, p12, p13], "c1"],
          [["spline3", p13, p14, p15], "c1"],
          [["spline3", p15, p16, p1], "c1"]]

for c, bc in curves:
    geo.Append(c, bc=bc, leftdomain=2, rightdomain=1)

# Add hold-all domain
geo.AddRectangle(p1=(0, 0),
                 p2=(1, 1),
                 bc="rectangle",
                 leftdomain=1,
                 rightdomain=0)

geo.SetMaterial(2, "w_int")
geo.SetMaterial(1, "w_out")

mesh = Mesh(geo.GenerateMesh(maxh=0.01))
mesh.Curve(3)

# Generate f to check domain in ParaView
X = CoefficientFunction((x, y))
f_values = {"w_int": 100, "w_out": -10}
f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])

#################################################################################################################
#                                               SECOND TEST                                                     #
#################################################################################################################

# Plot stuff
distPhi_u = []
numIterations = []

# We use the fully regularized state equation, which needs parameters c and γ.
for k in range(18):
    c = 2**k
    gamma = 500

    # Generate both obstacles
    phi1 = CoefficientFunction(0.5)
    phi2 = CoefficientFunction(5 * exp(-X[0] - 1))

    # Generate FES, test and trial functions
    fes = H1(mesh, order=1, dirichlet="rectangle")
    u = fes.TrialFunction()
    w = fes.TestFunction()

    # Generate max_γ and λ̄ where λ̄ can be chosen arbitrarily
    # lam_bar = CoefficientFunction(0)
    # lam_bar = CoefficientFunction((X[0] * (1 - X[0])) ** 2 * (X[1] * (1 - X[1])) ** 2 * 100)
    #
    # Convergence from the right direction is guaranteed for the following λ̄, which corresponds
    # to λ̄ = max(0, f + Δφ)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
    lam_bar = IfPos(-5 * exp(-X[0] - 1), -5 * exp(-X[0] - 1), 0)

    # Obstacle φ1 and φ2
    max_gam1 = IfPos(Norm(lam_bar + c * (u - phi2)) - 1 / gamma,
                     IfPos(lam_bar + c * (u - phi2),
                           lam_bar + c * (u - phi2),
                           0),
                     gamma / 4 * (lam_bar + c * (u - phi2)) * (lam_bar + c * (u - phi2)) + 1 / 2 * (
                                  lam_bar + c * (u - phi2)) + 1 / (4 * gamma))

    # Nonlinear variational formulation
    a1 = BilinearForm(fes, symmetric=False)
    a1 += grad(u) * grad(w) * dx
    a1 += -f * w * dx
    a1 += max_gam1 * w * dx

    u1 = GridFunction(fes)
    u1.vec[:] = 0

    u2 = GridFunction(fes)
    u2.vec[:] = 0

    # Solve fully regularized state equations
    gfu1 = GridFunction(fes)
    it = SimpleNewtonSolve(gfu1, a1)
    print(max(gfu1.vec[:]))
    mip = mesh(0.7, 0.5)
    distPhi_u.append([c, gfu1(mip) - phi2(mip)])
    numIterations.append([c, it])

np.savetxt('results/phi2_obs_dist', distPhi_u)
np.savetxt('results/phi2_obs_it', numIterations)

# Output in netget
Draw(f, mesh, "f")
Draw(gfu1, mesh, "gfu1")
Draw(phi1, mesh, "phi1")

# Output vtk
gff = GridFunction(fes)
gff.Set(f)

gfphi1 = GridFunction(fes)
gfphi1.Set(phi1)

# VTKOutput object
vtk = VTKOutput(ma=mesh,
                coefs=[gff, gfu1, gfphi1],
                names=["f", "u1", "phi"],
                filename="results/finalTarget",
                subdivision=1)
vtk.Do()

# Plotting stuff
distPhi_u_x, distPhi_u_y = zip(*distPhi_u)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 16,
})
plot1, = plt.semilogx(distPhi_u_x, distPhi_u_y)
plt.legend([plot1],[r"£\overline{\lambda} = 100 (x_1(1-x_1))^2 (x_2(1-x_2))^2£"], loc='upper right')
plt.title(r"Convergence of £\overline{\lambda}£")
plt.xlabel(r'c')
plt.ylabel(r'Signed distance between £y£ and £\varphi£')
plt.savefig("results/lam_barConvergence.svg")
```

A.3. Shape Derivative Formula

```
##########################################################################################################
#                                                                                                        #
# The aim of this program is to show convergence of (J(Ω_t) − J(Ω))/t → dJ(Ω)(θ) by doing a taylor test. #
#                                                                                                        #
##########################################################################################################

from auxiliaryFunctions import *
import matplotlib.pyplot as plt


############################################ *Generate target function* ############################################
maxH = 0.03
meshTarget = Mesh(getTargetMesh().GenerateMesh(maxh=maxH))
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
#meshTarget.Curve(4)

fesTarget = H1(meshTarget, order=3, dirichlet="rectangle")
uTarget = fesTarget.TrialFunction()
wTarget = fesTarget.TestFunction()

X = CoefficientFunction((x, y))
f_values = {"w_int": 100, "w_out": -10}
fTarget = CoefficientFunction([f_values[mat] for mat in meshTarget.GetMaterials()])
gffTarget = GridFunction(fesTarget)
gffTarget.Set(fTarget)

# Obstacle
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfphiTarget = GridFunction(fesTarget)
gfphiTarget.Set(phi)

# We have to solve the fully regularized problem
c = 1000
gam = 50
lam_barTarget = IfPos(fTarget, fTarget, 0)  # corresponds to lam_bar = max(0, f+\Delta \phi)
gflam_barTarget = GridFunction(fesTarget)
gflam_barTarget.Set(lam_barTarget)

# Nonlinear variational formulation
aTarget = BilinearForm(fesTarget, symmetric=False)
aTarget += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
aTarget += -fTarget * wTarget * dx(bonus_intorder=2)
aTarget += max_gam(uTarget, phi, c, gam, lam_barTarget) * wTarget * dx(bonus_intorder=2)

# Calculate target function
gfuTarget = GridFunction(fesTarget)
SimpleNewtonSolve(gfuTarget, aTarget, maxits=100)

########################################### *Calculate dJ(Ω)(θ)* ##########################################

# Set up initial stuff
mesh = Mesh(getStartMesh(maxH).GenerateMesh(maxh=maxH))
#mesh.Curve(4)
fes = H1(mesh, order=3, dirichlet="rectangle")
u = fes.TrialFunction()
w = fes.TestFunction()

f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])
gff = GridFunction(fes)
gff.Set(f)

lam_bar = IfPos(f, f, 0)
gflam_bar = GridFunction(fes)
gflam_bar.Set(lam_bar)

# Nonlinear variational formulation for state equation
a = BilinearForm(fes, symmetric=False)
a += grad(u) * grad(w) * dx(bonus_intorder=2)
a += -f * w * dx(bonus_intorder=2)
a += max_gam(u, phi, c, gam, lam_bar) * w * dx(bonus_intorder=2)

# Solve state equation
gfu = GridFunction(fes)
SimpleNewtonSolve(gfu, a, maxits=100)

# Nonlinear variational formulation for adjoint equation
p = fes.TrialFunction()
a_adj = BilinearForm(fes, symmetric=False)
a_adj += grad(p) * grad(w) * dx(bonus_intorder=2)
a_adj += (gfu - gfuTarget) * w * dx(bonus_intorder=2)
a_adj += c * sign_gam(gfu, phi, c, gam, lam_bar) * p * w * dx(bonus_intorder=2)

# Solve adjoint equation
gfp = GridFunction(fes)
SimpleNewtonSolve(gfp, a_adj, maxits=100)

# We want to compare results with ngsolve's DiffShape(), therefore we need the following extra steps
def Cost(u):
    return 1 / 2 * (u - gfuTarget) ** 2 * dx(bonus_intorder=2)


def StateEquation(u, w):
    return (grad(u) * grad(w) - f * w + max_gam(u, phi, c, gam, lam_bar) * w) * dx(bonus_intorder=2)


# Vector-valued Sobolev space
VEC = VectorH1(mesh, order=3, dirichlet="rectangle")
VT = VEC.TestFunction()
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
# Calculate dJ(\Omega)
LagrangeEquation = Cost(gfu) + StateEquation(gfu, gfp)
dJOmega = LinearForm(VEC)
dJOmega += LagrangeEquation.DiffShape(VT)
dJOmega.Assemble()

# Deformation vector field
theta = CoefficientFunction((10 * x * x * (1 - x) * (1 - x), 11 * y * y * (1 - y) * (1 - y)), dims=(2, 1))
gftheta = GridFunction(VEC)
gftheta.Set(theta)

# J(Ω)
JOmega = Integrate(Cost(gfu), mesh)

# dJ(\Omega)(\theta) using integral formula
dJOmega_theta_paper = Integrate(IntegrandPaper(gfu, phi, c, gam, lam_bar, gfuTarget, gfp, f, theta), mesh)

# dJ(\Omega)(\theta) using ngsolve's DiffShape() and adding missing term
dJOmega_theta_DiffShape = dJOmega(gftheta)
dJOmega_theta_DiffShape += Integrate(- (gfu - gfuTarget) * InnerProduct(grad(gfuTarget), theta), mesh)

################################################## *Taylor test* ##################################################

gfthetaTemp = GridFunction(VEC)
gfu_t = GridFunction(fes)
gfp_t = GridFunction(fes)

# Plot stuff
convShapeDiff = []
convPaper = []
prediction = []

# Taylor test
for n in range(20):
    # Calculate J(\Omega_t)
    gfthetaTemp.vec.data = 0.5 ** n * gftheta.vec
    mesh.SetDeformation(gfthetaTemp)
    SimpleNewtonSolve(gfu_t, a, maxits=100)
    JOmega_t = Integrate(Cost(gfu_t), mesh)
    mesh.UnsetDeformation()

    # Calculate J(\Omega_t) - J(\Omega) - t * dJ(\Omega)(\theta)
    convShapeDiff.append((0.5 ** n, abs(JOmega_t - JOmega - 0.5 ** n * dJOmega_theta_DiffShape)))
    convPaper.append((0.5 ** n, abs(JOmega_t - JOmega - 0.5 ** n * dJOmega_theta_paper)))

    # We calculate local error, therefore it has to be one order higher, i.e. O(t^2)
    prediction.append((0.5 ** n, (0.5 ** 2) ** n))

    # Debug stuff
    print(abs(JOmega_t - JOmega - 0.5 ** n * dJOmega_theta_DiffShape))
    print((JOmega_t - JOmega) / (0.5 ** n))
    print(dJOmega_theta_DiffShape)

# Plot stuff
convShapeDiff_x, convShapeDiff_y = zip(*convShapeDiff)
convPaper_x, convPaper_y = zip(*convPaper)
prediction_x, prediction_y = zip(*prediction)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.15, bottom=0.15)
plot1, plot2, plot3 = plt.loglog(convShapeDiff_x, convShapeDiff_y, convPaper_x, convPaper_y, prediction_x, prediction_y)
plt.legend([plot1, plot2, plot3], [r"DiffShape() corrected", r"Integral formula", r"£\mathcal{O}(t^2)£"], loc='upper left')
plt.xlabel(r't')
plt.ylabel(r'£|J(\Omega_t)-J(\Omega) - t^\textrm{d}J_{\gamma, c}(\Omega)(\theta)|£')
plt.title(r"Taylor Test")
plt.savefig("results/taylorTest.svg")
```

A.4. Shape Convergence

```
#################################################################################################################
#                                                                                                               #
# The aim of this program is to show convergence of the shape optimization problem as found in the paper      #
# Obstacle φ = 5 * exp(−x_1 − 1) and dJ(Ω) from ngSolve's DiffShape() and Integral formulation                 #
#                                                                                                               #
#################################################################################################################

from auxiliaryFunctions import *
import matplotlib.pyplot as plt
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
##################################################### *Setup* #####################################################

# Get target function
maxH = 0.02
X = CoefficientFunction((x, y))
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfuTarget = GetTargetFunction(maxH, phi)

# Set up initial stuff
mesh = Mesh(getStartMesh(maxH).GenerateMesh(maxh=maxH))
fes = H1(mesh, order=1, dirichlet="rectangle")
u = fes.TrialFunction()
w = fes.TestFunction()

f_values = {"w_int": 100, "w_out": -10}
f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])
gff = GridFunction(fes)
gff.Set(f)
lam_bar = IfPos(f, f, 0)
gflam_bar = GridFunction(fes)
gflam_bar.Set(lam_bar)

# Fully regularized problem
c = 10000
gam = 10000

# Nonlinear variational formulation for state equation
a_state = BilinearForm(fes, symmetric=False)
a_state += grad(u) * grad(w) * dx(bonus_intorder=2)
a_state += -f * w * dx(bonus_intorder=2)
a_state += max_gam(u, phi, c, gam, lam_bar) * w * dx(bonus_intorder=2)

# Solve state equation
gfu = GridFunction(fes)
SimpleNewtonSolve(gfu, a_state, maxits=100)

# Nonlinear variational formulation for adjoint equation
p = fes.TrialFunction()
a_adj = BilinearForm(fes, symmetric=False)
a_adj += grad(p) * grad(w) * dx(bonus_intorder=2)
a_adj += (gfu - gfuTarget)*w*dx(bonus_intorder=2)
a_adj += c*sign_gam(gfu, phi, c, gam, lam_bar) * p * w * dx(bonus_intorder=2)

# Solve adjoint equation
gfp = GridFunction(fes)
SimpleNewtonSolve(gfp, a_adj, maxits=100)

# We also want to compare with ngsolve's DiffShape(), therefore we need the following extra steps
def Cost(u):
    return 1/2*(u - gfuTarget) ** 2 * dx(bonus_intorder=2)


def StateEquation(u, w):
    return (grad(u)*grad(w) - f*w + max_gam(u, phi, c, gam, lam_bar)*w) * dx(bonus_intorder=2)


# Vector-valued Sobolev space
VEC = VectorH1(mesh, order=1, dirichlet="rectangle")
VT = VEC.TestFunction()

# Calculate dJ(\Omega) with corrected ngSolve's DiffShape() and the Integral formulation found in the paper
# Lagrange = Cost(gfu) + StateEquation(gfu, gfp)
# dJOmega_DiffShape = LinearForm(VEC)
# dJOmega_DiffShape += Lagrange.DiffShape(VT)
# dJOmega_DiffShape += - (gfu - gfuTarget) * InnerProduct(grad(gfuTarget), VT) * dx
dJOmega_paper = LinearForm(VEC)
dJOmega_paper += IntegrandPaper(gfu, phi, c, gam, lam_bar, gfuTarget, gfp, f, VT) * dx
dJOmega_paper += - (gfu - gfuTarget) * InnerProduct(grad(gfuTarget), VT) * dx\
                 - InnerProduct(grad(gfu), (grad(VT).trans + grad(VT)) * grad(gfp)) * dx\
                 - c * sign_gam(gfu, phi, c, gam, lam_bar) * InnerProduct(gradCF(phi), VT) * gfp * dx\
                 - InnerProduct(gradCF(f), VT) * gfp * dx\
                 + divVF(VT) * (1 / 2 * (gfu - gfuTarget) ** 2
                               + InnerProduct(grad(gfu), grad(gfp))
                               + max_gam(gfu, phi, c, gam, lam_bar) * gfp - f * gfp) * dx



def eps(u):
    return 1/2 * (grad(u)+grad(u).trans)

# Bilinear form for vector field
W, U = VEC.TnT()
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
aVEC = BilinearForm(VEC)
# aVEC += InnerProduct(grad(W), grad(U))*dx + InnerProduct(W, U)*dx
aVEC += InnerProduct(eps(W), eps(U)) * dx + InnerProduct(W, U) * dx
aVEC += 10*(grad(W)[1,1] - grad(W)[0,0])*(grad(U)[1,1] - grad(U)[0,0])*dx
aVEC += 10*(grad(W)[1,0] + grad(W)[0,1])*(grad(U)[1,0] + grad(U)[0,1])*dx


############################################## *Shape optimization* ##############################################

# Technical parameters
alpha = 0.5
alpha_incr_factor = 1.2
gamma_g = 1e-4
Nmax = 100
epsilon = 1e-7


gftheta = GridFunction(VEC)
gftheta.Set((0, 0))
gfVEC = GridFunction(VEC)
gfthetaTemp = GridFunction(VEC)

Jold = Integrate(Cost(gfu), mesh)

# For output
gfphi = GridFunction(fes)
gfphi.Set(phi)

isConverged = False

# Plot stuff
convDiffShape = []
firstRows = []

for k in range(Nmax):
    print(k)
    mesh.SetDeformation(gftheta)
    aVEC.Assemble()
    inv_aVEC = aVEC.mat.Inverse(VEC.FreeDofs(), inverse="sparsecholesky")
    # dJOmega_DiffShape.Assemble()
    dJOmega_paper.Assemble()
    # gfVEC.vec.data = inv_aVEC * dJOmega_DiffShape.vec
    gfVEC.vec.data = inv_aVEC * dJOmega_paper.vec
    # np.savetxt('results/firstRows_ngsolve_H3_order1', dJOmega_DiffShape.vec)
    np.savetxt('results/firstRows_paper_H3_order1', dJOmega_paper.vec)

    currentNormgfVEC = Norm(gfVEC.vec)

    while True:
        if currentNormgfVEC < epsilon:
            isConverged = True
            break
        gfthetaTemp.vec.data = gftheta.vec - alpha * gfVEC.vec
        mesh.SetDeformation(gfthetaTemp)
        gfphi = GridFunction(fes)
        gfphi.Set(phi)
        SimpleNewtonSolve(gfu, a_state, maxits=100)
        SimpleNewtonSolve(gfp, a_adj, maxits=100)
        Jnew = Integrate(Cost(gfu), mesh)
        mesh.UnsetDeformation()
        if Jnew < Jold - gamma_g * alpha*currentNormgfVEC**2:
            Jold = Jnew
            gftheta.vec.data = gfthetaTemp.vec
            alpha *= alpha_incr_factor
            print(alpha)
            print(Jnew)
            mesh.SetDeformation(gftheta)
            # VTKOutput object

            vtk = VTKOutput(ma=mesh,
                            coefs=[gfuTarget, gfu, gff, gfp, gfphi, gftheta],
                            names=["uTarget", "u", "f", "adj", "phi", "gfV"],
                            filename="results/shapeOptimization/resultOptimizationOpt_paper_H3_order1"+str(k),
                            subdivision=0)
            # Exporting the results:
            vtk.Do()
            break
        else:
            alpha = alpha / 2
            print(alpha)
            print(Jnew)
            print(Jold)
    convDiffShape.append([k, abs(Jnew)])
    np.savetxt('results/shapeConvergence_paper_H3_order1', convDiffShape)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
# Plot stuff
convDiffShape_x, convDiffShape_y = zip(*convDiffShape)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.15, bottom=0.15)
plot1, = plt.semilogy(convDiffShape_x, convDiffShape_y)
plt.legend([plot1], [r"£H_3£"], loc='upper right')
plt.xlabel(r'Iterations')
plt.ylabel(r'£J(\Omega)£')
plt.title(r"Shape optimization")
plt.savefig("results/ShapeOptimization_paper_H3_order1.svg")
```

A.5. Solving the Obstacle Problem

```
##########################################################################################################
#                                                                                                        #
# The aim of this program is to compare the regularized target function to the unregularized one        #
# 1st test: Obstacle φ = 5 * exp(−x_1 − 1)                                                                 #
#                                                                                                        #
##########################################################################################################
import numpy as np

from auxiliaryFunctions import *
import matplotlib.pyplot as plt

##################################################### *Setup* ##########################################

# Get target function
maxH = 0.05
meshTarget = Mesh(getTargetMesh().GenerateMesh(maxh=maxH))
#meshTarget.Curve(4)

fesTarget = H1(meshTarget, order=1, dirichlet="rectangle")
uTarget = fesTarget.TrialFunction()
wTarget = fesTarget.TestFunction()

X = CoefficientFunction((x, y))
f_values = {"w_int": 100, "w_out": -10}
fTarget = CoefficientFunction([f_values[mat] for mat in meshTarget.GetMaterials()])
gffTarget = GridFunction(fesTarget)
gffTarget.Set(fTarget)

# Obstacle
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfphiTarget = GridFunction(fesTarget)
gfphiTarget.Set(phi)

# Variational formulation for state equation
a_state = BilinearForm(fesTarget, symmetric=True)
a_state += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
a_state.Assemble()

b_state = LinearForm(fesTarget)
b_state += fTarget * wTarget * dx(bonus_intorder=2)
b_state.Assemble()

gfuUnregularizedTarget = GridFunction(fesTarget)
gfuUnregularizedTarget = getObstacleProblemSolution(a_state, b_state, gfphiTarget, meshTarget, fesTarget, kmax=450000)

diffTargets = []
for i in range(21):
    c = 2**i
    gam = 2**i
    # lam_barTarget = IfPos(fTarget, fTarget, 0)  # corresponds to lam_bar = max(0, f+\Delta \phi)
    lam_barTarget = IfPos(fTarget + 5 * exp(-X[0] - 1), fTarget + 5 * exp(-X[0] - 1),
                          0)  # corresponds to lam_bar = max(0, f+\Delta \phi)
    gflam_barTarget = GridFunction(fesTarget)
    gflam_barTarget.Set(lam_barTarget)

    # Nonlinear variational formulation
    aTarget = BilinearForm(fesTarget, symmetric=False)
    aTarget += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
    aTarget += -fTarget * wTarget * dx(bonus_intorder=2)
    aTarget += max_gam(uTarget, phi, c, gam, lam_barTarget) * wTarget * dx(bonus_intorder=2)

    # Calculate target function
    gfuRegularizedTarget = GridFunction(fesTarget)
    SimpleNewtonSolve(gfuRegularizedTarget, aTarget, maxits=100)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
    diffTargets.append([c, sqrt(Integrate((gfuUnregularizedTarget-gfuRegularizedTarget)**2 +
                                    (grad(gfuUnregularizedTarget)-grad(gfuRegularizedTarget))**2, meshTarget))])
    np.savetxt('results/diffTargets', diffTargets)
    print(diffTargets[i][1])



vtk = VTKOutput(ma=meshTarget,
                            coefs=[gfuRegularizedTarget, gfuUnregularizedTarget],
                            names=["gfuRegularizedTarget", "gfuUnregularizedTarget"],
                            filename="results/comparingTargets",
                            subdivision=0)
# Exporting the results:
vtk.Do()

print(Integrate((gfuUnregularizedTarget-gfuRegularizedTarget)**2, meshTarget))

# Plot stuff
diffTargets_x, diffTargets_y = zip(*diffTargets)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.20, bottom=0.15)
plot1, = plt.loglog(diffTargets_x, diffTargets_y)
plt.legend([plot1], [r"£\|y_\Omega - y_{\gamma, c, \Omega}\|_{H_0^1(D)}£"], loc='upper right')
plt.xlabel(r'£\gamma, c£')
plt.ylabel(r'£\|y_\Omega - y_{\gamma, c, \Omega}\|_{H_0^1(D)}£')
plt.title(r"Convergence of the obstacle problem")
plt.savefig("results/diffTargets.svg")
```

A.6. Calculating the Material Derivative

```
##########################################################################################################
#                                                                                                        #
# The aim of this program is to show convergence of (J(Ω_t) − J(Ω))/t → dJ(Ω)(θ) by doing a taylor      #
# test.                                                                                                  #
#                                                                                                        #
##########################################################################################################

from auxiliaryFunctions import *
import matplotlib.pyplot as plt

############################################ *Generate target function* ############################################
maxH = 0.01
meshTarget = Mesh(getTargetMesh().GenerateMesh(maxh=maxH))
#meshTarget.Curve(4)

fesTarget = H1(meshTarget, order=1, dirichlet="rectangle")
uTarget = fesTarget.TrialFunction()
wTarget = fesTarget.TestFunction()

X = CoefficientFunction((x, y))
f_values = {"w_int": 100, "w_out": -10}
fTarget = CoefficientFunction([f_values[mat] for mat in meshTarget.GetMaterials()])
gffTarget = GridFunction(fesTarget)
gffTarget.Set(fTarget)

# Obstacle
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfphiTarget = GridFunction(fesTarget)
gfphiTarget.Set(phi)

# Variational formulation for state equation
a_state = BilinearForm(fesTarget, symmetric=True)
a_state += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
a_state.Assemble()

b_state = LinearForm(fesTarget)
b_state += fTarget * wTarget * dx(bonus_intorder=2)
b_state.Assemble()

gfuTarget = getObstacleProblemSolution(a_state, b_state, gfphiTarget, meshTarget, fesTarget, kmax=450000)

########################################### *Calculate dJ(\Omega)(\theta)* ###########################################

# Set up initial stuff
mesh = Mesh(getStartMesh(maxH).GenerateMesh(maxh=maxH))
#mesh.Curve(4)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
fes = H1(mesh, order=1, dirichlet="rectangle")
u = fes.TrialFunction()
w = fes.TestFunction()

f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])
gff = GridFunction(fes)
gff.Set(f)


gfphi = GridFunction(fes)
gfphi.Set(phi)

# Vector-valued Sobolev space
VEC = VectorH1(mesh, order=3, dirichlet="rectangle")
VT = VEC.TestFunction()

# Deformation vector field
theta = CoefficientFunction((102 * x * x * (1 - x) * (1 - x) * y * y * (1 - y) * (1 - y),
                            231 * y * y * (1 - y) * (1 - y)* x * x * (1 - x) * (1 - x)), dims=(2, 1))
#theta = CoefficientFunction((-x,0), dims=(2, 1))
gftheta = GridFunction(VEC)
gftheta.Set(theta)

# Variational formulation for state equation
a_state = BilinearForm(fes, symmetric=True)
a_state += grad(u) * grad(w) * dx(bonus_intorder=2)
a_state.Assemble()

b_state = LinearForm(fes)
b_state += gff * w * dx(bonus_intorder=2)
b_state.Assemble()

# Solve state equation
gfstate = getObstacleProblemSolution(a_state, b_state, gfphi, mesh, fes)

# Determine indices for S_u(K)
nabla = BilinearForm(fes, symmetric=False)
nabla += grad(u) * grad(w) * dx(bonus_intorder=2)
nabla.Assemble()
rows,cols,vals = nabla.mat.COO()
NABLA = sp.sparse.csr_matrix((vals,(rows,cols)))

mass = BilinearForm(fes, symmetric=False)
mass += u * w * dx(bonus_intorder=2)
mass.Assemble()
rows,cols,vals = mass.mat.COO()
MASS = sp.sparse.csr_matrix((vals,(rows,cols)))

indices = []
uNP = gfstate.vec.FV().NumPy()[:]
fNP = gff.vec.FV().NumPy()[:]
result1 = NABLA*uNP
result2 = MASS*fNP

for k in range(len(result1)):
    if(abs(result1[k]-result2[k]) < 1e-4):
        indices.append(k)
        print(k)


# Solve material derivative equation
b_mat = LinearForm(fes)
b_mat += - InnerProduct(grad(gfstate) ,(- grad(gftheta).trans - grad(gftheta))  * grad(w)) * dx
b_mat += - div(gftheta) * grad(gfstate) * grad(w) * dx
b_mat += div(gftheta) * gff * w * dx
b_mat.Assemble()

gfmat = getMaterialDerivativeProblemSolution(a_state, b_mat, gfphi, gfstate, indices, mesh, fes)

# The cost function
def Cost(u):
    return 1 / 2 * (u - gfuTarget) ** 2 * dx(bonus_intorder=2)


# J(Ω)
JOmega = Integrate(Cost(gfstate), mesh)

dJOmega_unregularized = Integrate(-(gfstate - gfuTarget) * InnerProduct(grad(gfuTarget), gftheta) + 1/2 *
                                  (gfstate - gfuTarget)**2 * div(gftheta) + (gfstate - gfuTarget) * gfmat, mesh)

################################################## *Taylor test* ##################################################

gfthetaTemp = GridFunction(VEC)
gfu_t = GridFunction(fes)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
# Plot stuff
convUnreg = []
prediction = []

# Taylor test
for n in range(14):
    # Calculate J(\Omega_t)
    gfthetaTemp.vec.data = 0.5 ** n * gftheta.vec
    mesh.SetDeformation(gfthetaTemp)
    # Variational formulation for state equation
    a_state = BilinearForm(fes, symmetric=True)
    a_state += grad(u) * grad(w) * dx(bonus_intorder=2)
    a_state.Assemble()

    b_state = LinearForm(fes)
    b_state += gff * w * dx(bonus_intorder=2)
    b_state.Assemble()
    # SimpleNewtonSolve(gfu_t, a, maxits=100)
    gfstate_t = getObstacleProblemSolution(a_state, b_state, gfphi, mesh, fes)
    JOmega_t = Integrate(Cost(gfstate_t), mesh)
    mesh.UnsetDeformation()

    # Calculate J(Ω_t) − J(Ω) − t ∗ dJ(Ω)(θ)
    convUnreg.append((0.5 ** n, abs(JOmega_t - JOmega - 0.5 ** n * dJOmega_unregularized)))

    # We calculate local error, therefore it has to be one order higher, i.e. O(t^2)
    prediction.append((0.5 ** n, (0.5 ** 2) ** n))

    # Debug stuff
    print((JOmega_t - JOmega) / (0.5 ** n))
    print(dJOmega_unregularized)

# Plot stuff
convUnreg_x, convUnreg_y = zip(*convUnreg)
prediction_x, prediction_y = zip(*prediction)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.15, bottom=0.15)
plot1, plot2= plt.loglog(convUnreg_x, convUnreg_y, prediction_x, prediction_y)
plt.legend([plot1, plot2], [r"Unregularized Shape Derivative", r"£\mathcal{O}(t^2)£"], loc='upper left')
plt.xlabel(r't')
plt.ylabel(r'£|J(\Omega_t)-J(\Omega) - t^\textrm{d}J_{\gamma, c}(\Omega)(\theta)|£')
plt.title(r"Taylor Test for the Unregularized Problem")
plt.savefig("results/taylorTestUnregularized.svg")
```

A.7. Solving the Shape Optimization Problem

```
##########################################################################################################
#
# The aim of this program is to show convergence of the unregularized shape optimization problem
#
##########################################################################################################

from auxiliaryFunctions import *
import matplotlib.pyplot as plt

##################################################### *Setup* ##################################################

# Get target function
maxH = 0.02
meshTarget = Mesh(getTargetMesh().GenerateMesh(maxh=maxH))
#meshTarget.Curve(4)

fesTarget = H1(meshTarget, order=1, dirichlet="rectangle")
uTarget = fesTarget.TrialFunction()
wTarget = fesTarget.TestFunction()

X = CoefficientFunction((x, y))
f_values = {"w_int": 100, "w_out": -10}
fTarget = CoefficientFunction([f_values[mat] for mat in meshTarget.GetMaterials()])
gffTarget = GridFunction(fesTarget)
gffTarget.Set(fTarget)

# Obstacle
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfphiTarget = GridFunction(fesTarget)
gfphiTarget.Set(phi)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
# Variational formulation for state equation
a_state = BilinearForm(fesTarget, symmetric=True)
a_state += grad(uTarget) * grad(wTarget) * dx(bonus_intorder=2)
a_state.Assemble()

b_state = LinearForm(fesTarget)
b_state += fTarget * wTarget * dx(bonus_intorder=2)
b_state.Assemble()

gfuTarget = getObstacleProblemSolution(a_state, b_state, gfphiTarget, meshTarget, fesTarget, kmax=450000)

################################################ *   More Setup   * ################################################

# Set up initial stuff
mesh = Mesh(getStartMesh(maxH).GenerateMesh(maxh=maxH))
fes = H1(mesh, order=1, dirichlet="rectangle")
u = fes.TrialFunction()
w = fes.TestFunction()

gfphi = GridFunction(fes)
gfphi.Set(phi)

f_values = {"w_int": 100, "w_out": -10}
f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])
gff = GridFunction(fes)
gff.Set(f)

# Variational formulation for state equation
a_state = BilinearForm(fes, symmetric=True)
a_state += grad(u) * grad(w) * dx(bonus_intorder=2)
a_state.Assemble()

b_state = LinearForm(fes)
b_state += gff * w * dx(bonus_intorder=2)
b_state.Assemble()

# Solve state equation
gfu = getObstacleProblemSolution(a_state, b_state, gfphi, mesh, fes)

# Determine indices for S_u(K)
nabla = BilinearForm(fes, symmetric=False)
nabla += grad(u) * grad(w) * dx(bonus_intorder=2)
nabla.Assemble()
rows,cols,vals = nabla.mat.COO()
NABLA = sp.sparse.csr_matrix((vals,(rows,cols)))

mass = BilinearForm(fes, symmetric=False)
mass += u * w * dx(bonus_intorder=2)
mass.Assemble()
rows,cols,vals = mass.mat.COO()
MASS = sp.sparse.csr_matrix((vals,(rows,cols)))

indices = []
uNP = gfu.vec.FV().NumPy()[:]
fNP = gff.vec.FV().NumPy()[:]
result1 = NABLA*uNP
result2 = MASS*fNP

for k in range(len(result1)):
    if(abs(result1[k]-result2[k]) < 1e-4):
        indices.append(k)
        print(k)

# Vector-valued Sobolev space
VEC = VectorH1(mesh, order=1, dirichlet="rectangle")
VT = VEC.TestFunction()

gftheta = GridFunction(VEC)
gftheta.Set((0, 0))
gfVEC = GridFunction(VEC)
gfthetaTemp = GridFunction(VEC)

# Solve material derivative equation
b_mat = LinearForm(fes)
b_mat += - InnerProduct(grad(gfu) ,(- grad(gftheta).trans - grad(gftheta))  * grad(w)) * dx
b_mat += - div(gftheta) * grad(gfu) * grad(w) * dx
b_mat += div(gftheta) * gff * w * dx

gfmat = getMaterialDerivativeProblemSolution(a_state, b_mat, gfphi, gfu, indices, mesh, fes)

# Calculate dJ(\Omega)
dJOmega_Unregularized = LinearForm(VEC)
dJOmega_Unregularized += -(gfu-gfuTarget)*InnerProduct(grad(gfuTarget), VT) * dx
dJOmega_Unregularized += 1 / 2 * (gfu-gfuTarget)**2 * divVF(VT) * dx
dJOmega_Unregularized += (gfu-gfuTarget) * gfmat * dx
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
def eps(u):
    return 1/2 * (grad(u)+grad(u).trans)

# Bilinear form for vector field
W, U = VEC.TnT()
aVEC = BilinearForm(VEC)
# aVEC += InnerProduct(grad(W), grad(U))*dx + InnerProduct(W, U)*dx
aVEC += InnerProduct(eps(W), eps(U)) * dx + InnerProduct(W, U) * dx
aVEC += 10*(grad(W)[1,1] - grad(W)[0,0])*(grad(U)[1,1] - grad(U)[0,0])*dx
aVEC += 10*(grad(W)[1,0] + grad(W)[0,1])*(grad(U)[1,0] + grad(U)[0,1])*dx


################################################ *Shape optimization* ################################################

# Technical parameters
alpha = 0.5
alpha_incr_factor = 1.2
gamma_g = 1e-4
Nmax = 100
epsilon = 1e-7


# The cost function
def Cost(u):
    return 1/2*(u - gfuTarget) ** 2 * dx(bonus_intorder=2)


Jold = Integrate(Cost(gfu), mesh)

# For output
gfphi = GridFunction(fes)
gfphi.Set(phi)

isConverged = False

# Plot stuff
convDiffShape = []
firstRows = []

for k in range(Nmax):
    print(k)
    mesh.SetDeformation(gftheta)
    aVEC.Assemble()
    inv_aVEC = aVEC.mat.Inverse(VEC.FreeDofs(), inverse="sparsecholesky")
    dJOmega_Unregularized = LinearForm(VEC)
    dJOmega_Unregularized += -(gfu - gfuTarget) * InnerProduct(grad(gfuTarget), VT) * dx
    dJOmega_Unregularized += 1 / 2 * (gfu - gfuTarget) ** 2 * divVF(VT) * dx
    dJOmega_Unregularized += (gfu - gfuTarget) * gfmat * dx
    dJOmega_Unregularized.Assemble()
    gfVEC.vec.data = inv_aVEC * dJOmega_Unregularized.vec

    currentNormgfVEC = Norm(gfVEC.vec)

    while True:
        if currentNormgfVEC < epsilon:
            isConverged = True
            break

        gfthetaTemp.vec.data = gftheta.vec - alpha * gfVEC.vec
        mesh.SetDeformation(gfthetaTemp)
        gfphi.Set(phi)

        a_state.Assemble()
        # Solve material derivative equation
        b_mat = LinearForm(fes)
        b_mat += - InnerProduct(grad(gfu), (- grad(gftheta).trans - grad(gftheta)) * grad(w)) * dx
        b_mat += - div(gftheta) * grad(gfu) * grad(w) * dx
        b_mat += div(gftheta) * gff * w * dx
        b_state.Assemble()
        # Solve state equation
        gfu = getObstacleProblemSolution(a_state, b_state, gfphi, mesh, fes)

        indices = []
        uNP = gfu.vec.FV().NumPy()[:]
        fNP = gff.vec.FV().NumPy()[:]
        result1 = NABLA * uNP
        result2 = MASS * fNP

        indices = []
        for i in range(len(result1)):
            if (abs(result1[i] - result2[i]) < 1e-3):
                indices.append(i)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
        b_mat.Assemble()
        gfmat = getMaterialDerivativeProblemSolution(a_state, b_mat, gfphi, gfu, indices, mesh, fes)

        Jnew = Integrate(Cost(gfu), mesh)
        mesh.UnsetDeformation()
        gfphi.Set(phi)
        if Jnew < Jold - gamma_g * alpha*currentNormgfVEC**2:
            Jold = Jnew
            gftheta.vec.data = gfthetaTemp.vec
            alpha *= alpha_incr_factor
            print(alpha)
            print(Jnew)
            mesh.SetDeformation(gftheta)
            gfphi.Set(phi)
            # VTKOutput object

            vtk = VTKOutput(ma=mesh,
                            coefs=[gfuTarget, gfu, gff, gfphi, gftheta],
                            names=["uTarget", "u", "f", "phi", "gfV"],
                            filename="results/shapeOptimization/resultOptimizationOpt_unregularized_H3_"+str(k),
                            subdivision=0)
            # Exporting the results:
            vtk.Do()
            break
        else:
            alpha = alpha / 2
            print(alpha)
            print(Jnew)
            print(Jold)
    convDiffShape.append([k, abs(Jnew)])
    np.savetxt('results/shapeConvergence_unregularized_H3', convDiffShape)


#lam_bar0 = np.loadtxt('results/lam_bar0')
#lam_bar1 = np.loadtxt('results/lam_bar1')
# Plot stuff
convDiffShape_x, convDiffShape_y = zip(*convDiffShape)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.15, bottom=0.15)
plot1, = plt.semilogy(convDiffShape_x, convDiffShape_y)
plt.legend([plot1], [r"£H_3£"], loc='upper right')
plt.xlabel(r'Iterations')
plt.ylabel(r'£J(\Omega)£')
plt.title(r"Shape optimization")
plt.savefig("results/ShapeOptimization_Unregularized_H3.svg")
```

A.8. Comparing the Regularized with the Unregularized Problem

```
#######################################################################################################
#                                                                                                     #
# The aim of this program is to demonstrate the mixed approach with the material derivative of the regularized problem #
#                                                                                                     #
#######################################################################################################

from auxiliaryFunctions import *
import matplotlib.pyplot as plt

##################################################### *Setup* ##################################################

# Get target function
maxH = 0.02
X = CoefficientFunction((x, y))
# phi = CoefficientFunction(0.5)
phi = CoefficientFunction(5 * exp(-X[0] - 1))
gfuTarget = GetTargetFunction(maxH, phi)

# Set up initial stuff
mesh = Mesh(getStartMesh(maxH).GenerateMesh(maxh=maxH))
fes = H1(mesh, order=1, dirichlet="rectangle")
u = fes.TrialFunction()
w = fes.TestFunction()

f_values = {"w_int": 100, "w_out": -10}
f = CoefficientFunction([f_values[mat] for mat in mesh.GetMaterials()])
gff = GridFunction(fes)
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
gff.Set(f)
lam_bar = IfPos(f, f, 0)
gflam_bar = GridFunction(fes)
gflam_bar.Set(lam_bar)

# Fully regularized problem
c = 10000
gam = 10000

# Variational formulation for state equation
a_state = BilinearForm(fes, symmetric=False)
a_state += grad(u) * grad(w) * dx(bonus_intorder=2)
a_state += -f * w * dx(bonus_intorder=2)
a_state += max_gam(u, phi, c, gam, lam_bar) * w * dx(bonus_intorder=2)

# Solve state equation
gfu = GridFunction(fes)
SimpleNewtonSolve(gfu, a_state, maxits=100)


def Cost(u):
    return 1/2*(u - gfuTarget) ** 2 * dx(bonus_intorder=2)


def StateEquation(u, w):
    return (grad(u)*grad(w) - f*w + max_gam(u, phi, c, gam, lam_bar)*w) * dx(bonus_intorder=2)


# Vector-valued Sobolev space
VEC = VectorH1(mesh, order=1, dirichlet="rectangle")
VT = VEC.TestFunction()

gftheta = GridFunction(VEC)
gftheta.Set((0, 0))
gfVEC = GridFunction(VEC)
gfthetaTemp = GridFunction(VEC)

a_regMat = BilinearForm(fes, symmetric=False)
a_regMat += grad(u) * grad(w) * dx(bonus_intorder=2)
a_regMat += div(gftheta) * grad(gfu) * grad(w) * dx(bonus_intorder=2)
a_regMat += InnerProduct(grad(gfu) ,(- grad(gftheta).trans - grad(gftheta))  * grad(w)) * dx(bonus_intorder=2)
a_regMat += c * sign_gam(gfu, phi, c, gam, lam_bar) * u * w * dx(bonus_intorder=2)
a_regMat += div(gftheta) * max_gam(gfu, phi, c, gam, lam_bar) * w * dx(bonus_intorder=2)
a_regMat += -f * div(gftheta) * w * dx(bonus_intorder=2)

# Solve state equation
gf_regMat = GridFunction(fes)
SimpleNewtonSolve(gf_regMat, a_regMat, maxits=100)

dJOmega_regularized_Mat = Integrate(-(gfu - gfuTarget) * InnerProduct(grad(gfuTarget), gftheta) +
                                    1/2 *(gfu - gfuTarget)**2 * div(gftheta) + (gfu - gfuTarget) * gf_regMat, mesh)


def eps(u):
    return 1/2 * (grad(u)+grad(u).trans)

# Bilinear form for vector field
W, U = VEC.TnT()
aVEC = BilinearForm(VEC)
# aVEC += InnerProduct(grad(W), grad(U))*dx + InnerProduct(W, U)*dx
aVEC += InnerProduct(eps(W), eps(U)) * dx + InnerProduct(W, U) * dx
aVEC += 10*(grad(W)[1,1] - grad(W)[0,0])*(grad(U)[1,1] - grad(U)[0,0])*dx
aVEC += 10*(grad(W)[1,0] + grad(W)[0,1])*(grad(U)[1,0] + grad(U)[0,1])*dx


############################################## *Shape optimization* ##############################################

# Technical parameters
alpha = 0.5
alpha_incr_factor = 1.2
gamma_g = 1e-4
Nmax = 100
epsilon = 1e-7


Jold = Integrate(Cost(gfu), mesh)

# For output
gfphi = GridFunction(fes)
gfphi.Set(phi)

isConverged = False

# Plot stuff
convDiffShape = []
```


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
firstRows = []

for k in range(Nmax):
    print(k)
    mesh.SetDeformation(gftheta)
    aVEC.Assemble()
    inv_aVEC = aVEC.mat.Inverse(VEC.FreeDofs(), inverse="sparsecholesky")
    dJOmega_Unregularized = LinearForm(VEC)
    dJOmega_Unregularized += -(gfu - gfuTarget) * InnerProduct(grad(gfuTarget), VT) * dx
    dJOmega_Unregularized += 1 / 2 * (gfu - gfuTarget) ** 2 * divVF(VT) * dx
    dJOmega_Unregularized += (gfu - gfuTarget) * gf_regMat * dx
    dJOmega_Unregularized.Assemble()
    gfVEC.vec.data = inv_aVEC * dJOmega_Unregularized.vec

    currentNormgfVEC = Norm(gfVEC.vec)

    while True:
        if currentNormgfVEC < epsilon:
            isConverged = True
            break

        gfthetaTemp.vec.data = gftheta.vec - alpha * gfVEC.vec
        mesh.SetDeformation(gfthetaTemp)
        gfphi.Set(phi)

        SimpleNewtonSolve(gfu, a_state, maxits=100)
        SimpleNewtonSolve(gf_regMat, a_regMat, maxits=100)

        Jnew = Integrate(Cost(gfu), mesh)
        mesh.UnsetDeformation()
        gfphi.Set(phi)
        if Jnew < Jold - gamma_g * alpha*currentNormgfVEC**2:
            Jold = Jnew
            gftheta.vec.data = gfthetaTemp.vec
            alpha *= alpha_incr_factor
            print(alpha)
            print(Jnew)
            mesh.SetDeformation(gftheta)
            gfphi.Set(phi)
            # VTKOutput object

            vtk = VTKOutput(ma=mesh,
                            coefs=[gfuTarget, gfu, gff, gfphi, gftheta],
                            names=["uTarget", "u", "f", "phi", "gfV"],
                            filename="results/shapeOptimization/resultOptimizationOpt_regularizedMAT_H3_"+str(k),
                            subdivision=0)
            # Exporting the results:
            vtk.Do()
            break
        else:
            alpha = alpha / 2
            print(alpha)
            print(Jnew)
            print(Jold)
    convDiffShape.append([k, abs(Jnew)])
    np.savetxt('results/shapeConvergence_regularizedMAT_H3_order1', convDiffShape)

# Plot stuff
convDiffShape_x, convDiffShape_y = zip(*convDiffShape)
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern"],
    "font.size": 15,
})
plt.gcf().subplots_adjust(left=0.15, bottom=0.15)
plot1, = plt.semilogy(convDiffShape_x, convDiffShape_y)
plt.legend([plot1], [r"£H_3£"], loc='upper right')
plt.xlabel(r'Iterations')
plt.ylabel(r'£J(\Omega)£')
plt.title(r"Shape optimization")
plt.savefig("results/ShapeOptimization_regularizedMAT_H3_order1.svg")
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

[1] H. Amann and J. Escher. *Analysis II*. Springer-Verlag, Berlin Heidelberg New York, 2006.

[2] A. Arnold. *Variationsrechnung*. TU Wien, Lecture Notes, March 2018.

[3] D. Bucur and G. Buttazzo. *Variational Methods in Shape Optimization Problems*. Springer Science & Business Media, Berlin Heidelberg, 2006.

[4] C. Christof, C. Meyer, S. Walther, and C. Christian. Optimal Control of a Non-Smooth Semilinear Elliptic Equation. *Mathematical Control & Related Fields*, 8(1):247–276, 2018.

[5] M. C. Delfour and J.-P. Zolesio. *Shapes and Geometries - Metrics, Analysis, Differential Calculus, and Optimization, Second Edition*. Society for Industrial & Applied Mathematics (SIAM), Philadelphia, 2011.

[6] Z. Dostál. *Optimal Quadratic Programming Algorithms - With Applications to Variational Inequalities*. Springer Science & Business Media, Berlin Heidelberg, 2009.

[7] L. C. Evans. *Partial Differential Equations*. American Mathematical Soc., Heidelberg, 2010.

[8] D. Fremlin. *Kirszbraun's Theorem*. University of Essex, Colchester, England, 2018.

[9] B. Führ, V. Schulz, and K. Welker. Shape Optimization for Interface Identification with Obstacle Problems. *Vietnam Journal of Mathematics*, 46(4):967–985, Oct. 2018.

[10] P. Gangl, K. Sturm, M. Neunteufel, and J. Schöberl. Fully and Semi-Automated Shape Differentiation in NGSolve. *Structural and Multidisciplinary Optimization*, 63(3):1579–1607, Nov. 2020.

[11] C. Heinemann and K. Sturm. Shape Optimization for a Class of Semilinear Variational Inequalities with Applications to Damage Models. *SIAM Journal on Mathematical Analysis*, 48(5):3579–3617, Jan. 2016.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[12] M. Hintermüller and A. Laurain. Optimal Shape Design Subject to Elliptic Variational Inequalities. *SIAM Journal on Control and Optimization*, 49(3):1015–1047, Jan. 2011.

[13] K. Ito and K. Kunisch. Semi–Smooth Newton Methods for Variational Inequalities of the First Kind. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(1):41–62, Jan. 2003.

[14] M. Kaltenbäck. *Analysis 3 für technische Mathematik*. TU Wien, Lecture Notes, 2015.

[15] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. Classics in Applied Mathematics. Society for Industrial & Applied Mathematics (SIAM), 1980.

[16] A. Laurain, M. Winckler, and I. Yousept. Shape Optimization for Superconductors Governed by H(curl)-Elliptic Variational Inequalities. *SIAM Journal on Control and Optimization*, 59(3):2247–2272, Jan. 2021.

[17] D. Luft, V. H. Schulz, and K. Welker. Efficient Techniques for Shape Optimization with Variational Inequalities Using Adjoints. *SIAM Journal on Optimization*, 30(3):1922–1953, Jan 2020.

[18] M. Růžička. *Nichtlineare Funktionalanalysis - Eine Einführung*. Springer-Verlag, Berlin, Heidelberg, New York, 2004.

[19] J. Sokołowski and J. Zolésio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer-Verlag, Berlin, 1992.

[20] K. Sturm. *On Shape Optimization with Non-Linear Partial Differential Equations*. PhD thesis, Technische Universität Berlin, 2015.

[21] K. Sturm. Shape Differentiability Under Non-linear PDE Constraints. In *New Trends in Shape Optimization*, page 271–300. Springer International Publishing, 2015.

[22] K. Sturm. *Lecture on Shape Optimization*. TU Wien, Lecture Notes, 2019.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN