Efficient 3D Shape Optimization with Isogeometric Analysis

Limkilde, Asger

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Limkilde, A. (2020). Efficient 3D Shape Optimization with Isogeometric Analysis. Technical University of Denmark.

[^0]Department of Applied Mathematics and Computer Science

# Efficient 3D Shape Optimization with Isogeometric Analysis 

## Asger Limkilde

Kongens Lyngby 2020

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 45253031
compute@compute.dtu.dk
www.compute.dtu.dk

## Summary

The subject of this thesis is the development of methods to efficiently solve 3D shape optimization problems with isogeometric analysis. Shape optimization is the art of finding the best possible shape based on a desired property. Such problems often arise in engineering, where the criterion depends on the solution to a partial differential equation, that models the underlying physical process.

Isogeometric analysis is a numerical method for solving partial differential equations posed on a complicated domain. With this method the shape can be represented exactly, by using splines, and the optimization can be performed directly on parameters controlling the shape of the domain.

When using isogeometric analysis for shape optimization one needs to maintain a valid parametrization of the interior of the computational domain during the optimization process. This is not a trivial task, and often methods rely on non-linear constraints on the validity of the parametrization. When considering 3D shape optimization problems the number of constraints can be huge.

In this work we will investigate a regularization based approach to maintaining parametrizations, that avoids the expensive constraints. As the first contribution we demonstrate that this approach performs at par with a constraint based method, when considering a 2D model problem of designing electromagnetic reflectors.

As the second contribution we use this regularization based shape optimization approach to find parametrizations. We demonstrate that the method is able to find parametrizations of complicated shapes both in 2D and 3D, and the result serves as a benchmark that illustrates the capability of the method.

Finally as the third contribution we consider a 3D shape optimization problem of designing reflectors for free surface waves. With the proposed approach we are able to obtain a shape that performs much better than the initial guess, while maintaining a valid parametrization. However to obtain a valid parametrization the optimization had to be terminated prematurely.

## Resumé (in Danish)

Denne afhandling beskriver udviklingen af effektive metoder til at løse 3D form optimerings problemer med isogeometrisk analyse. Form optimering er kunsten at finde den bedst mulige form med hensyn til en bestem egenskab. Sådanne problemer opstår ofte i ingeniør videnskab, hvor den $ø$ nskede egenskab afhænger af løsningen til en partiel differentialligning, der modellerer den underliggende fysiske proces.

Isogeometrisk analyse er en numerisk metode til at løse partielle differential ligninger på domæner med en kompliceret form. Med denne metode kan formen repræsenteres eksakt, med brug af splines, og optimering kan dermed blive udført direkte på parametre der styrer denne form.

Når isogeometrisk analyse bruges til form optimering, har man brug for at vedligeholde en valid, dvs. bijektiv, parametrisering af det indre af det beregningsmæssige domæne i løbet af optimerings processen. Dette er ikke en triviel opgave og ofte afhænger metoder af ikke-lineære bibetingelser på validiteten af parametriseringen. Antallet af bibetingelser kan være meget stort når man betragter 3D form optimerings problemer.

I dette studie undersøger vi en metode til at vedligeholde parametriseringer, der baserer sig på regularisering, og undgår disse dyre bibetingelser. Som det første bidrag demonstrerer vi at denne metode præsterer sammenligneligt med en metode baseret på bibetingelser, ved at betragte et 2D model problem med at designe elektromagnetiske reflektorer.

Som det andet bidrag, bruger vi denne regularisering baserede form optimerings metode til at finde parametriseringer. Vi demonstrerer at denne metode kan finde parametriseringer af kompliserede former både i 2D og 3D, og resultatet tjener som et benchmark der illustrerer kapaciteten af metoden.

Til slut, som det tredje bidrag, betragter vi et 3D form optimerings problem med at designe reflektorer for fri-overflade bølger. Med den forslåede metode er vi i stand til at finde former der præsterer mange gange bedre end start gættet, mens en valid parametrisering er vedligeholdt. For at opnå en valid parametrisering måtte optimeringen imidlertidigt standes for tidligt.

## Preface

This thesis is submitted in partial fulfillment of the requirements for obtaining the degree of Ph.D. at the Technical University of Denmark (DTU). The project was funded by DTU and were carried out at the department of Applied Mathematics and Computer Science (DTU Compute) from the 1st of September 2017 to the 31st of August 2020. The Ph.D. project was supervised by associate professor Jens Gravesen from DTU Compute and associate professor Anton Evgrafov from Department of Mathematical Sciences at Aalborg University. A 5 months external stay was conducted at the Norwegian University of Science and Technology with Anton Evgrafov (who were employed there at the time) and a 5 weeks external stay was conducted at the AROMATH group at Inria Sophia Antipolis-Méditerranée Research Centre with Angelos Mantzaflaris.

During the three years of the Ph.D. study the following manuscript has been written

- Limkilde, A., Evgrafov, A., and Gravesen, J., Mantzaflaris A. "Practical isogeometric shape optimization: Parameterization by means of regularization", submitted to the Journal of Computational Design and Engineering, 2020.

Part on the content of the thesis, specifically Chapter 3, rest on this manuscript, and it is included as an appendix. Some of the figures Chapter 3 are taken directly from this manuscript. However the thesis is written as a self contained text.

The intended reader of the thesis has an engineering background with some basic knowledge on Galerkin methods, for example the finite element method, Sobolev spaces and optimization.

I have done the implementations in C++, using the G+Smo library for the building blocks of the isogeometric analysis, and the optimization library Ipopt for the optimization algorithms. The code can be found at https://github.com/gismo/ shapeopt.

## Acknowledgements

First I would like to thank my supervisors Jens Gravesen and Anton Evgrafov for our many fruitful discussions and their excellent advice and suggestions. Next I would like to thank the people at AROMATH for being very welcoming during my research stay with them, and I would like to thank Angelos Mantzaflaris for introducing me
to the G+Smo library and for sharing his insights on how to effectively implement isogeometric analysis and maintaining code in general. I would also like to thank my Ph.D. colleagues DTU compute for creating a good work environment. Finally I want to thank Amalie for her loving support.

Kongens Lyngby, August 31, 2020
Asjus himlilule

## Contents

Summary ..... i
Resumé (in Danish) ..... iii
Preface ..... v
Contents ..... vii
Notation and Abbreviations ..... ix
Introduction ..... xi
1 Preliminaries ..... 1
1.1 Splines and B-splines ..... 1
1.2 Isogeometric Analysis ..... 3
1.3 Integration by Quadrature ..... 6
1.4 Multiple Patches ..... 6
1.5 Computing Derivatives ..... 7
2 Parametrization Techniques ..... 11
2.1 Linear Parametrization Techniques ..... 12
2.2 Validity Constraints ..... 14
2.3 Optimization Based Methods ..... 17
2.4 Examples ..... 20
3 2D Shape Optimization using IGA ..... 27
3.1 Shape Optimization using Linearizations ..... 28
3.2 Shape Optimization using Regularization ..... 33
3.3 Optimization of Electromagnetic Reflectors ..... 36
3.4 Comparison and Discussion ..... 48
4 Isogeometric Parameterizations using Shape Optimization Methods ..... 51
4.1 Finding an Initial Guess ..... 52
4.2 Snapping Strategies ..... 53
4.3 Choice of Regularization Parameter ..... 54
4.4 Scaling ..... 54
4.5 Multipatch Strategies ..... 54
4.6 2D Examples ..... 58
4.7 3D Examples ..... 65
4.8 Conclusion ..... 69
5 3D shape optimization with IGA ..... 71
5.1 Problem Description ..... 71
5.2 Shape Optimization Framework ..... 78
5.3 Results ..... 79
5.4 Discussion ..... 89
5.5 Conclusion and outlook ..... 90
6 Conclusion and outlook ..... 93
6.1 Conclusion ..... 93
6.2 Outlook ..... 94
A Paper: "Practical isogeometric shape optimization: Parameterization by means of reg- ularization" ..... 95
B Results using Regularizations ..... 115
C Perfectly Matched Layers ..... 121
Bibliography ..... 125

## Notation and Abbreviations

$\Xi$ : The knot vector
$\xi_{i}$ : The $i$ th knot
$p$ : The degree of a spline
$n$ : The dimension of a spline space
$C^{r}$ : Continuously differentiable r times
$\mathcal{S}_{\Xi}^{p}$ : The spline space with knotvector $\Xi$ and degree $p$
$\mathcal{N}_{i}$ : The $i$ th B-spline
$d$ : The dimension, typically $d=2$ or $d=3$
$\mathcal{R}_{i}$ : The $i$ th $d$-variate tensor product B -spline
$\boldsymbol{\Xi}$ : A d-tuple of knot vectors, $\boldsymbol{\Xi}=\left(\Xi_{1}, \ldots, \Xi_{d}\right)$
$\boldsymbol{p}$ : A d-tuple of degrees, $\boldsymbol{p}=\left(p_{1}, \ldots, p_{d}\right)$
$\mathcal{S}_{\boldsymbol{\Xi}}^{\boldsymbol{p}}$ : The tensor product spline space given by $\boldsymbol{\Xi}$ and $\boldsymbol{p}$
$\mathcal{S}$ : A spline space or tensor product spline space
$\boldsymbol{\xi}:$ A point in the parameter domain, $\boldsymbol{\xi} \in \mathbb{R}^{d}$
$\xi^{(i)}$ : The $i$ th coordinate of a point in the parameter domain
$\boldsymbol{x}$ : A point in the physical domain, $\boldsymbol{x} \in \mathbb{R}^{d}$
$c$ : The vector of the coordinates of the control points
$c^{B}$ : The vector of the coordinates of the boundary control points
$\boldsymbol{c}^{\boldsymbol{I}}$ : The vector of the coordinates of the inner control points
$W$ : The Winslow functional
$G:$ The geometry map (also called the parametrization)
$\mathcal{S}^{g}$ : The spline space of $G$
$J$ : The Jacobian matrix of $G$
$\mathcal{S}_{\text {det }}$ : The spline space of $\operatorname{det} J$
$\widehat{\mathcal{R}}_{i}$ : The $i$ th basis spline of $\mathcal{S}_{\text {det }}$
$\boldsymbol{d}$ : The vector of spline coefficients of $\operatorname{det} J$
$\boldsymbol{d}^{(k)}$ : The vector of spline coefficients of det $J$ when expanded in its spline space uniformly refined k times

2D : 2-dimensional Euclidean space
3D : 3-dimensional Euclidean space
IGA : Isogeometric Analysis
CAD : Computer Aided Design
FEA : Finite Element Analysis
FEM : Finite Element Method
PDE : Partial Differential Equation

## Introduction

Shape optimization problems are problems where the goal is to find the best possible shape with respect to certain criteria. As shape optimization problems most often arise from engineering applications, the problems often involve simulations of physical processes, for example heat dissipation [23], electromagnetism [33], fluid dynamics [36], etc. Such physical processes are governed by a Partial Differential Equation (PDE) posed on the domain which shape is to be optimized. The coupling between the representation of the shape and the simulations are of great importance in shape optimization, and many strategies has been developed. Examples include topology optimization [4] where one represent the shape as a material distribution within a design domain and level set methods [2] where the shape is represented as the level set of a function defined on a design domain. However, designing in engineering are commonly done using Computer Aided Design (CAD) software. Here the geometry is represented using B-splines or NURBS (non uniform rational B-splines). This poses the requirement on shape optimization framework that one should be able to convert the result into the CAD software for further design or production. However this is in many cases not at trivial task.

Isogeometric Analysis (IGA) is a relatively new framework for numerical solution of PDEs, which was proposed in 2005 [25]. The method was developed with the aim of bridging the gap between design and simulation It shares features with Finite Element Analysis (FEA) as both methods are Galerkin methods [12, 46] but the main difference is that in FEA the computational domain is approximated by a polyhedral mesh, while in IGA the computational domain is parametrized using splines. This makes IGA very appealing for shape optimization, as the optimization can be perform directly on parameters controlling the spline parametrization, namely its so called control points. This has the potential that the shape optimization algorithm produces results that can be readily imported into CAD software. In IGA one also uses splines as the approximation space for the Galerkin method, which has the advantage that an approximation space of arbitrarily high degree of smoothness can be constructed, which for smooth PDEs increases the accuracy per degree of freedom [11].

One of the main challenges in IGA in general is that the shape of a domain only depends on its boundary, but in IGA one needs a parametrization of the interior of the domain. The quality of this parametrization can affect the accuracy of the simulations $[20,44]$ and at the very least it should be a bijective map, e.g., its Jacobian determinant should be positive, in which case we call the parametrization 'valid'.

There are some simple methods for generating a parametrization of the interior for example the Coons patch method [14] and the spring method [20]. These methods depend linearly on the boundary control points, however they fail to produce a valid parametrization for more complicated domains. To be able to handle more complex domains several nonlinear optimization based methods has been proposed [20, 43]. Notably the minimization of the Winslow functional [20, 42] has proved to be effective for finding 2D parametrizations, and this functional will be used multiple times in this work. Finally some recent development in this field includes a PDE-based approach [22] and an approach based on elastic deformations [39]. The challenge of constructing parametrizations is especially important when using IGA for shape optimization, as a valid parametrization has to be maintained during an optimization process where the shape of the domain changes.

IGA has been successfully applied to shape optimization problems in 2D in many studies. For example for vibrating membranes [29], for problems in electromagnetic and electrical systems [34, 17] and in fluid dynamics [36]. Often methods rely on constraints on the spline coefficients of the Jacobian determinant, to ensure that the parametrization is valid, for example [34, 17]. These constraints are sometimes too strict, in which case it is necessary to relax them by refining the spline space in which the Jacobian determinant is expanded. This can lead to an excessive number of constraints, especially in 3D where in addition to the extra dimension, the Jacobian determinant also has higher degree. A new promising attempt to avoid these constraints have been proposed in [23] where the geometry parametrization enters the formulation as an additional PDE constraint. However it is not clear whether this approach generalizes to 3 D problems. Another approach to shape optimization that avoids constraints on the Jacobian determinant is where one optimizes over all the control points of the parametrization (not only the ones affecting the boundary) and adds a regularization term to the objective function that drives the optimization towards parametrizations of good quality. This method has been investigated for shape optimization with FEM in [37] and has, to my knowledge, only been investigated very briefly in the context of shape optimization with IGA in [15].

There has been relative few studies that consider shape optimization problems in 3D with IGA. In [28, 7] a Boundary Element Method (BEM) is used for 3D shape optimization using IGA with T-splines. In BEM the PDE is reduced to to an integral formulation on the boundary, and thus a parametrization of the interior is not needed. The disadvantage of BEM is that even though the system matrices are smaller, they are not sparse and generally not symmetric. Additionally the Greens function for the PDE has to be known to be able to use BEM, which is not always the case. Other methods heavily constraint the geometry, for example by only allowing deformations along a radial direction [24].

In this work the overall objective is to develop methods for efficient 3D shape optimization. In particular we will consider the challenge of maintaining valid parametrizations during the optimization process, in a way that allows for large deformations.

We will first consider a method based on using linearizations of the Winslow minimization problem for constructing parametrizations, coupled with constraints on the Jacobian determinant, similar to the method in [34]. The new aspect in this work is that we use adaptive refinement to relax the constraints on the coefficients of the Jacobian determinant where needed, and hence reduce the number of constraints compared to uniform refinement. We will compare this method to a regularization based method, similar to the one in [15], on a 2D model problem of designing electromagnetic reflectors. This method avoids explicit constraints on the Jacobian determinant, however it only guarantees that the Jacobian determinant is positive in the quadrature points.

Next we consider a new application of shape optimization, by formulating the parametrization challenge as a shape optimization problem. We demonstrate how the regularization based shape optimization approach can be used to find parametrizations of complicated domains in 2D and 3D, avoiding the explicit constraints on the Jacobian determinant.

Finally we investigate the regularization based approach on a 3D shape optimization problem of designing reflectors for free surface flow. The structure of the thesis is as follows.

## Structure

- In Chapter 1 we introduce splines and IGA. The goal of this chapter is to set the scene and fix the notation for the rest of the thesis.
- In Chapter 2 we consider some methods for finding parametrizations in IGA, that will serve as benchmarks.
- In Chapter 3 we introduce the two shape optimization frameworks considered in this work. We will compare the two methods on a 2D problem of designing electromagnetic reflectors.
- In Chapter 4 we use the regularization based shape optimization approach to find parametrizations in 2D and 3D. We demonstrate that this is an effective technique for finding parametrizations in its own right. However this application will also serve as benchmark for shape optimization problems, that demonstrate the capabilities of the method.
- In Chapter 5 we investigate the regularization based approach on the 3D shape optimization problem of designing reflectors for free surface flow.
- Finally, in Chapter 6 we conclude and consider possible directions for future work.

The methods are implemented for multi patch geometries using the IGA library $\mathrm{G}+\mathrm{Smo}^{1}{ }^{1}[30]$. For performing the optimization we use the optimization library IpOpt ${ }^{2}$ [41].

[^1]
## сhapter 1

## Preliminaries

In this chapter we will give an overview of splines, B-splines and Isogeometric analysis for discretizing PDE's. The aim of the chapter is to set the scene and fix the notation for the rest of the thesis. For a more thorough introduction to IGA we refer the reader to [11].

### 1.1 Splines and B-splines

Given a degree $p \in \mathbb{R}$, an integer $n \in \mathbb{N}$ and a set of points $\xi_{i} \in \mathbb{R}$ for $i=1, \ldots, n+p+1$ a spline is a function $\gamma:\left[\xi_{p+1}, \xi_{n+1}[\rightarrow \mathbb{R}\right.$ which is a polynomial on each of the intervals [ $\xi_{i}, \xi_{i+1}\left[\right.$ for $i=p+1, \ldots, n$. The points $\xi_{i}$ is called the knots, and together they form the knotvector $\Xi=\left(\xi_{1}, \ldots, \xi_{n+p+1}\right)$. The knots $\xi_{p+1}, \ldots, \xi_{n+1}$ will be referred to as the inner knots. A spline is $C^{p-k}$ at a knot with multiplicity $k$, meaning that the knot is repeated $k$ times. We will in this work consider open knotvectors, i.e., where the first and last knot has multiplicity $p+1$. Given a knotvector $\Xi$ and a degree $p$ we will let $\mathcal{S}_{\Xi}^{p}$ denote the space of splines with this degree and knotvector.

A particular nice basis for the spline space $\mathcal{S}_{\Xi}^{p}$ is the so called B-splines $\mathcal{N}_{i}$. They can be constructed by the Cox de boor algorithm, for details see [11]. An example of B-splines are shown in Figure 1.1.

Theorem 1. For $B$-splines, $\mathcal{N}_{i}(\xi)$ for $i=1, \ldots, n$, of degree $p$ with open knotvector $\Xi$ the following properties holds

- They are non-negative: $\mathcal{N}_{i}(\xi) \geq 0$
- They form a partition of unity: $\sum_{i}^{n} \mathcal{N}_{i}(\xi)=1$
- The restriction of a $B$-spline $\mathcal{N}_{i}$ to an open $\left.k n o t ~ i n t e r v a l ~\right] ~ \xi_{j}, \xi_{j+1}[$ is a polynomial of degree $p$.

A spline curve in $\mathbb{R}^{2}$ is a curve $\gamma=(x, y)$ such that $x \in \mathcal{S}_{\Xi}^{p}$ and $y \in \mathcal{S}_{\Xi}^{p}$. It can be written in terms of the B-splines as

$$
\begin{equation*}
\gamma(\xi)=\sum_{i}^{n} \boldsymbol{c}_{i} \mathcal{N}_{i}(\xi) \tag{1.1}
\end{equation*}
$$



Figure 1.1: B Splines of degree $p$, with inner knots $(0,0.25,0.5,0.75,1)$


Figure 1.2: A spline curve of degree $p=2$, with inner knots $(0,0.25,0.5,0.75,1)$
where $\boldsymbol{c}_{i}=\left(c_{i}^{x}, c_{i}^{y}\right)$. The points $\boldsymbol{c}_{i}$ are called the control points and together they form the control polygon. An example of a spline curve is shown in Figure 1.2.

Until now we have considered the univariate case. For the multivariate case we will consider tensor product splines. We let $d$ denote the dimension. Given $d$ knotvectors $\boldsymbol{\Xi}=\left(\Xi_{1}, \ldots, \Xi_{d}\right)$ and $d$ degrees $\boldsymbol{p}=\left(p_{1}, \ldots, p_{d}\right)$ the $d$-variate tensor product spline is a function $\left.\left.\left.r:] \xi_{p_{1}+1}^{(1)}, \xi_{n_{1}+1}^{(1)}\right] \times \cdots \times\right] \xi_{p_{d}+1}^{(d)}, \xi_{n_{d}+1}^{(d)}\right] \rightarrow \mathbb{R}$. Such that if the arguments $\xi^{(1)}, \ldots, \xi^{(i-1)}, \xi^{(i+1)}, \ldots, \xi^{(d)}$ are fixed then $r(\boldsymbol{\xi})$ is a spline, where $\boldsymbol{\xi}=\left(\xi^{(i)}, \ldots, \xi^{(d)}\right)$. We will denote the tensor product splines space with knotvectors $\boldsymbol{\Xi}$ and degrees $\boldsymbol{p}$ by
$\mathcal{S}_{\underline{\Xi}}^{p}$. A basis for this space is the tensor product B-splines given as

$$
\begin{equation*}
\mathcal{R}_{j}(\boldsymbol{\xi})=\mathcal{R}_{i_{1}, \ldots, i_{d}}\left(\xi^{(1)}, \ldots, \xi^{(d)}\right)=\mathcal{N}_{i_{1}}^{(1)}\left(\xi^{(1)}\right) \cdot \cdots \cdot \mathcal{N}_{i_{d}}^{(d)}\left(\xi^{(d)}\right) \tag{1.2}
\end{equation*}
$$

where $j$ is a global index. We will in this work only consider the cases $d=2$ and $d=3$, also referred to as 2D and 3D.

A d-dimensional spline surface can be written as

$$
\begin{equation*}
r(\boldsymbol{\xi})=\sum_{i=1}^{n} \boldsymbol{c}_{i} \mathcal{R}_{i}(\boldsymbol{\xi}) \tag{1.3}
\end{equation*}
$$

Here we misuse the notation a bit, and now let $n$ denote the number of tensor product B-splines. An example of a spline surface in 2D is shown in Figure 1.3.


Figure 1.3: Spline surface of degree $p=2$, with inner knots $(0,1 / 3,1 / 6,1)$ in both directions, and its control polygon. The grey lines are parameter lines, i.e., the image of a uniform grid.

### 1.2 Isogeometric Analysis

In this section we will introduce IGA for approximating solutions to PDEs. We will use a Helmholtz equation as an example, as this is the type of PDE that we shall see later for our shape optimization examples. We will consider the Helmholtz PDE with Neumann boundary conditions.

$$
\begin{align*}
\hat{\Delta} \hat{u}+K \hat{u} & =0 & & \text { in } \Omega  \tag{1.4a}\\
\frac{\partial \hat{u}}{\partial \boldsymbol{n}} & =g & & \text { on } \Gamma_{N}  \tag{1.4b}\\
\frac{\partial \hat{u}}{\partial \boldsymbol{n}} & =0 & & \text { on } \Gamma_{0} \tag{1.4c}
\end{align*}
$$



Figure 1.4: Sketch of the parametrization and boundary conditions

IGA is a Galerkin method and the first step is to write (1.4) in the weak formulation. This is done by multiplying (1.4) with a test function $\hat{v} \in H^{1}(\Omega)$ and integrate over the domain $\Omega$ :

$$
\begin{equation*}
\int_{\Omega} \hat{\Delta} \hat{u} \cdot \hat{v} \mathrm{~d} x+K \int_{\Omega} \hat{u} \hat{v} \mathrm{~d} x=0 \tag{1.5}
\end{equation*}
$$

Now we can use Greens identity to move one of the derivatives in the first term from $\hat{u}$ to $\hat{v}$ yielding

$$
\begin{equation*}
-\int_{\Omega}\langle\hat{\nabla} \hat{u}, \hat{\nabla} \hat{v}\rangle_{2} \mathrm{~d} x+\int_{\partial \Omega} \frac{\partial \hat{u}}{\partial \boldsymbol{n}} \hat{v} \mathrm{~d} s+K \int_{\Omega} \hat{u} \hat{v} \mathrm{~d} x=0 \tag{1.6}
\end{equation*}
$$

where $\langle\cdot, \cdot\rangle_{2}$ is the Euclidian inner product. Using the boundary conditions (1.4b) and (1.4c) we arrive at the weak formulation: Find $\hat{u} \in H^{1}(\Omega)$ such that

$$
\begin{equation*}
\int_{\Omega}\langle\hat{\nabla} \hat{u}, \hat{\nabla} \hat{v}\rangle_{2} \mathrm{~d} \hat{V}-K \int_{\Omega} \hat{u} \hat{v} \mathrm{~d} \hat{V}=\int_{\Gamma_{s}} g \hat{v} \mathrm{~d} \hat{S} \tag{1.7}
\end{equation*}
$$

holds for all test functions $\hat{v} \in H^{1}(\Omega)$.
With IGA we now consider a parametrization of the physical domain $G:[0,1]^{d} \rightarrow$ $\Omega$ such that $G\left([0,1]^{d}\right)=\Omega$, and where $d$ is the dimension. It is assumed that the parameter domain is the unit square $[0,1]^{d}$ for simplicity. We need the parametrization to be bijective, i.e., $\operatorname{det} J \neq 0$, and we will in this work consider the case where the Jacobian determinant is to be positive $\operatorname{det} J>0$. To discretize this equation we will use splines both for representing the geometry and for approximating $u$. We will look for a solution in a tensor product spline space which we will denote by $\mathcal{S}$, we will consider $G \in \mathcal{S}^{g}$, that is

$$
\begin{equation*}
G=\sum_{i}^{n^{g}} c_{i} \mathcal{R}_{i}^{g} \tag{1.8}
\end{equation*}
$$

The superscript $g$ indicates that the spline space for the geometry map can be different from the one we will use to approximate $u$. The setup is sketched in Figure 1.4.

We have that

$$
\hat{\nabla}=J^{-T} \nabla
$$

where $\hat{\nabla}$ are the gradient in the physical domain, such that $\hat{\nabla}_{i}=\frac{\partial}{\partial x^{(i)}}$ and $\nabla$ is the gradient in the parameter domain, such that $\nabla_{i}=\frac{\partial}{\partial \xi^{(i)}}$. The Jacobian matrix of $G$ is denoted by $J$. We now pull back the equation (1.7) to the parameter domain to get the weak formulation: Find $u=\hat{u} \circ G$ such that

$$
\begin{array}{r}
\int_{[0,1]^{d}}\left\langle J^{-T} \nabla u, J^{-T} \nabla v\right\rangle_{2} \operatorname{det} J \mathrm{~d} V-K \int_{[0,1]^{d}} u v \operatorname{det} J \mathrm{~d} V  \tag{1.9}\\
=\int_{G^{-1}\left(\Gamma_{s}\right)} g \circ G v\left\|\frac{\partial G}{\partial \nu}\right\| \mathrm{d} S,
\end{array}
$$

for all test functions $v=\hat{v} \circ G$. Here $\nu$ is the boundary variable. Note that the measure $\left\|\frac{\partial G}{\partial \nu}\right\|$ is for the case $d=2$. For $d=3$ we get $\left\|\frac{\partial G}{\partial \nu_{1}} \times \frac{\partial G}{\partial \nu_{2}}\right\|$. We will introduce the bilinear form

$$
\begin{equation*}
a(u, v)=a_{1}(u, v)-a_{2}(u, v) \tag{1.10}
\end{equation*}
$$

where $a_{1}$ and $a_{2}$ are bilinear forms given by

$$
\begin{align*}
& a_{1}(u, v)=\int_{[0,1]^{d}}\left\langle J^{-T} \nabla u, J^{-T} \nabla v\right\rangle_{2} \operatorname{det} J \mathrm{~d} V  \tag{1.11a}\\
& a_{2}(u, v)=K \int_{[0,1]^{d}} u v \operatorname{det} J \mathrm{~d} V \tag{1.11b}
\end{align*}
$$

Finally we define the linear form

$$
\begin{equation*}
\ell(v)=\int_{G^{-1}\left(\Gamma_{s}\right)} g \circ G v\left\|\frac{\partial G}{\partial \nu}\right\| \mathrm{d} S, \tag{1.12}
\end{equation*}
$$

We look for $u_{h} \in \mathcal{S}$

$$
\begin{equation*}
u_{h}=\sum_{i}^{n} u_{i} \mathcal{R}_{i} \tag{1.13}
\end{equation*}
$$

such that (3.17) holds for all test functions $v \in \mathcal{S}$. Since $\mathcal{S}$ is a vector space it is sufficient to test against the basis functions $\mathcal{R}_{i}$. This yields the system of linear equations

$$
\begin{equation*}
A \boldsymbol{u}=B \boldsymbol{u}+C \boldsymbol{u}=\boldsymbol{F} \tag{1.14}
\end{equation*}
$$

where $\boldsymbol{u}=\left(u_{1}, \ldots, u_{n}\right)^{T}$ and

$$
\begin{align*}
B_{i j} & =\int_{[0,1]^{d}}\left\langle J^{-T} \nabla \mathcal{R}_{i}, J^{-T} \nabla \mathcal{R}_{j}\right\rangle_{2} \operatorname{det} J \mathrm{~d} V,  \tag{1.15a}\\
C_{i j} & =K \int_{[0,1]^{d}} \mathcal{R}_{i} \mathcal{R}_{j} \operatorname{det} J \mathrm{~d} V,  \tag{1.15b}\\
F_{j} & =\int_{G^{-1}\left(\Gamma_{s}\right)} g \circ G \mathcal{R}_{j}\left\|\frac{\partial G}{\partial \nu}\right\| \mathrm{d} S . \tag{1.15c}
\end{align*}
$$

We have not yet discussed how to choose $G$ given boundary curves that defines the shape of the computational domain, as this will be the topic of the next chapter. This challenge will be revisited throughout the thesis, as it is one of the main challenges when using IGA in general, and especially when using IGA for shape optimization.

### 1.3 Integration by Quadrature

To compute the integrals in (1.15), we will, unless otherwise specified, use GaussLegendre quadrature. We will refer to this as Gauss quadrature. Here an integral is approximated by a linear combination of function evaluations

$$
\begin{equation*}
\int_{[-1,1]} f(t) \mathrm{d} t \approx \sum_{i}^{n_{Q}} w_{i} f\left(t_{i}\right) \tag{1.16}
\end{equation*}
$$

where $w_{i}$ is referred to as the weights and $t_{i}$ is referred to as the Gauss abscissas, or quadrature points. We will refer to this quadrature rule simply as Gauss quadrature. For further details we refer the reader to [6]. The integral above is over the interval $[-1,1]$ however integration over an arbitrary interval $[a, b]$ can be accomplished by the change of variables $s=\frac{(1-t) a+(1+t) b}{2}$. Throughout this work we will perform the Gauss quadrature over each knot interval whenever the integrand involves splines, as for example in (1.15). Gauss quadrature has the property that integrals of polynomials of degree $2 n_{Q}-1$ can be computed exactly with this method. This means that some of the integrals we saw in the last section, for example the one in (1.11b) can be computed exactly. Other integrals, for example the one in (1.11a) has an integrand which is a rational polynomial. In this case we will use the number of quadrature points necessary to exactly integrate the denominator multiplied by the numerator, unless stated otherwise.

The Gauss-Legendre quadrature is an open quadrature rule, meaning that it does not include function evaluations at the end points of the interval. But for some applications, as we shall see later in Chapter 4 and 5, it can be advantageous to include function evaluations at the end points. Such a quadrature rule is called a closed quadrature rule and an example that we will use in this work is the GaussLobatto quadrature. The Gauss-Lobatto quadrature integrates polynomials of degree $2 n_{Q}-3$ exactly when using $n_{Q}$ quadrature points.

### 1.4 Multiple Patches

When considering more complicated topologies of the physical domain, one might not able to parametrize the domain using one spline parametrization. In this case one splits the domain into multiple patches, such that each patch can be parametrized using splines. The interface where two patches meet are referred to as a patch interface. We will in this work also use multiple patches where material properties change. This
makes it possible to change shape of parts of the domain with a specific material property, by changing the patch interfaces.

For each patch $p$ we will have a parametrization $G_{p}$, which means that the weak formulation in the parameter domain will depend on the patch $p$. However in most cases the weak parametrization can be written such that the only difference between patches is the parametrization. Due to this we will give the weak formulations in general form, in terms of $G$, even when multiple patches are used.

We enforce $C^{0}$ continuity of the solution at the patch interfaces. This is accomplished in G+Smo by identifying a two connected controlpoints along the interface between two patches as a single entity. For further details we refer the reader to [11, chapter 3].

### 1.5 Computing Derivatives

When using IGA for shape optimization we need to be able to differentiate an objective function that depends on $u_{h}$ with respect to the coordinates of the control points. We will illustrate our approach for doing this in this section. We consider a function

$$
\begin{equation*}
E\left(u_{h}, \boldsymbol{c}\right)=\int_{\Omega} e\left(u_{h}, \boldsymbol{c}\right) \mathrm{d} \boldsymbol{\xi} \tag{1.17}
\end{equation*}
$$

We will in this work treat $u_{h}$ as a function of $\boldsymbol{c}$. We can calculate the derivative of $E$ a with the chain rule

$$
\begin{equation*}
\frac{\partial E}{\partial \boldsymbol{c}}\left(u_{h}, \boldsymbol{c}\right)=\frac{\partial E}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}+\frac{\partial E}{\partial \boldsymbol{c}}, \tag{1.18}
\end{equation*}
$$

where $\boldsymbol{u}$ is the spline coefficients of $u_{h}$. We will here focus on the first term $\frac{\partial E}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}$, as this will illustrate the main ideas. By differentiation of (1.14) we find that the term $\frac{\partial u}{\partial c}$ has to satisfy

$$
\begin{equation*}
A \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}=\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{c}}-\frac{\partial A}{\partial \boldsymbol{c}} \boldsymbol{u} \tag{1.19}
\end{equation*}
$$

But to avoid solving this system of equations that has $d n^{g}$ right hand sides we use the adjoint method. Here we rewrite

$$
\begin{equation*}
\frac{\partial E}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}=\frac{\partial E}{\partial \boldsymbol{u}} A^{-1}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{c}}-\frac{\partial A}{\partial \boldsymbol{c}} \boldsymbol{u}\right)=\left(\frac{\partial E}{\partial \boldsymbol{u}} A^{-1}\right)\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{c}}-\frac{\partial A}{\partial \boldsymbol{c}} \boldsymbol{u}\right) . \tag{1.20}
\end{equation*}
$$

Now we can compute $\frac{\partial E}{\partial \boldsymbol{u}} A^{-1}$ which is a linear system with only one right hand side. Next we demonstrate how to compute the terms $\frac{\partial A}{\partial c} \boldsymbol{u}$ and $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{c}}$. For the term $\frac{\partial A}{\partial \boldsymbol{c}} \boldsymbol{u}$ we will use that the $i$ th element of $A \boldsymbol{u}$ can be written as $[A \boldsymbol{u}]_{i}=a\left(\mathcal{R}_{i}, u_{h}\right)$. As an example we can differentiate $a_{1}\left(\mathcal{R}_{i}, u_{h}\right)$ with respect to a coordinate of a controlpoint
as

$$
\begin{align*}
\frac{\partial a_{1}}{\partial c_{j}}\left(\mathcal{R}_{i}, u_{h}\right) & =\int_{[0,1]^{d}}\left\langle\frac{\partial J^{-T}}{\partial c_{j}} \nabla \mathcal{R}_{i}, J^{-T} \nabla u_{h}\right\rangle_{2} \operatorname{det} J \mathrm{~d} V \\
& +\int_{[0,1]^{d}}\left\langle J^{-T} \nabla \mathcal{R}_{i}, \frac{\partial J^{-T}}{\partial c_{j}} \nabla u_{h}\right\rangle_{2} \operatorname{det} J \mathrm{~d} V  \tag{1.21}\\
& +\int_{[0,1]^{d}}\left\langle J^{-T} \nabla \mathcal{R}_{i}, J^{-T} \nabla u_{h}\right\rangle_{2} \frac{\partial \operatorname{det} J}{\partial c_{j}} \mathrm{~d} V
\end{align*}
$$

here we can use the formulas

$$
\begin{align*}
\frac{\partial J^{-1}}{\partial c_{j}} & =-J^{-1} \frac{\partial J}{\partial c_{j}} J^{-1}  \tag{1.22a}\\
\frac{\partial \operatorname{det} J}{\partial c_{j}} & =\operatorname{det} J \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial c_{j}}\right) . \tag{1.22b}
\end{align*}
$$

We have that

$$
\begin{equation*}
G=\sum_{j}^{d n^{g}} c_{j} \boldsymbol{\mathcal { R }}_{j}^{g} \tag{1.23}
\end{equation*}
$$

where

$$
\boldsymbol{\mathcal { R }}_{j}^{g}=\boldsymbol{R}_{i+(k-1) \cdot n^{g}}^{g}=\mathcal{R}_{i}^{g} \boldsymbol{e}_{k}=\left[\begin{array}{c}
0  \tag{1.24}\\
\vdots \\
\mathcal{R}_{i}^{g} \\
\vdots \\
0
\end{array}\right]
$$

which means that

$$
\begin{align*}
& \frac{\partial G}{\partial c_{j}}=\boldsymbol{\mathcal { R }}_{j}^{g},  \tag{1.25a}\\
& \frac{\partial J}{\partial c_{j}}=J_{j}, \tag{1.25b}
\end{align*}
$$

where $J$ is the Jacobian matrix of $G$ and $J_{j}$ is the Jacobian matrix of $\boldsymbol{\mathcal { R }}_{j}^{g}$. This means that we can write the derivative of $a_{1}\left(\mathcal{R}_{i}, u_{h}\right)$ with respect to $c_{j}$ as a bilinear form $d a_{u_{h}}^{1}$ applied to basis functions $\mathcal{R}_{i}$ and $\boldsymbol{\mathcal { R }}_{j}^{g}$ :

$$
\begin{equation*}
\frac{\partial[B \boldsymbol{u}]_{i}}{\partial c_{j}}=\frac{\partial a_{1}}{\partial c_{j}}\left(\mathcal{R}_{i}, u_{h}\right)=d a_{u_{h}}^{1}\left(\mathcal{R}_{i}, \boldsymbol{\mathcal { R }}_{j}^{g}\right), \tag{1.26}
\end{equation*}
$$

where $B$ is the matrix arising from $a_{1}$ as defined in (1.15a). with

$$
\begin{align*}
d a_{u_{h}}^{1}\left(\mathcal{R}_{i}, \boldsymbol{\mathcal { R }}_{j}^{g}\right)= & -\int_{[0,1]^{d}}\left\langle J^{-T} J_{j}^{T} J^{-T} \nabla \mathcal{R}_{i}, J^{-T} \nabla u_{h}\right\rangle_{2} \operatorname{det} J \mathrm{~d} V \\
& -\int_{[0,1]^{d}}\left\langle J^{-T} \nabla \mathcal{R}_{i}, J^{-T} J_{j}^{T} J^{-T} \nabla u_{h}\right\rangle_{2} \operatorname{det} J \mathrm{~d} V  \tag{1.27}\\
+ & \int_{[0,1]^{d}}\left\langle J^{-T} \nabla \mathcal{R}_{i}, J^{-T} \nabla u_{h}\right\rangle_{2} \operatorname{det} J \operatorname{tr}\left(J^{-1} J_{j}\right) \mathrm{d} V .
\end{align*}
$$

This formulation is nice since we use the G+Smo library for the implementations, and this library has efficient implementations for the assembly of matrices and vectors arising from bilinear and linear forms, like the one above. The same approach can be used for computing the derivatives of $C \boldsymbol{u}$. When computing the derivative $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{c}}$ the approach is also the same except that the measure in the integral is different as it is an integral over the boundary. This measure can be written as

$$
\begin{equation*}
\left\|\frac{\partial G}{\partial \nu_{1}} \times \frac{\partial G}{\partial \nu_{2}}\right\|=\sqrt{\sum_{j=1} d \operatorname{det} M_{i j}^{2}} \tag{1.28}
\end{equation*}
$$

where $M_{i j}$ is the $i, j$ th minor of the Jacobian, i.e., where the $i$ th row and the $j$ th column are removed from $J$. Here $i$ is the index of the parameter that are fixed on this boundary. This expression is nice since it is dimension independent, for example for $d=2$ the equality holds:

$$
\begin{equation*}
\left\|\frac{\partial G}{\partial \nu}\right\|=\sqrt{\left(J_{1 j}\right)^{2}+\left(J_{2 j}\right)^{2}}=\sqrt{\sum_{j=1} d \operatorname{det} M_{i j}^{2}} \tag{1.29}
\end{equation*}
$$

where $J_{i j}$ are the entry at $i, j$. This expression can be differentiated with respect to $c_{j}$ using that the minors $M_{i j}$ are linearly dependent on $J$, together with (1.25b) and the rules given in (1.22).

# CHAPTER <br> 2 

## Parametrization Techniques

One of the main building blocks of IGA is the geometry parametrization of the physical domain. As described in the last chapter it is used to pull back the weak formulation of a PDE before discretization. But while the parametrization has to parametrize the entire domain, the shape of the domain is given only by its boundary. This means that we need to be able find a parameterization of the interior of the domain, given the boundary curves.

In IGA the parametrization is represented by a set of control points $\boldsymbol{c}$. These can either be on the boundary or in the interior. Typically the boundary control points will be fixed, but this could also be the case for some of the inner control points. In this chapter will in the following use the notation

$$
c=\left[\begin{array}{l}
c^{I}  \tag{2.1}\\
c^{B}
\end{array}\right]
$$

where $\boldsymbol{c}^{\boldsymbol{I}}$ are the free control points, typically the inner control points, and $\boldsymbol{c}^{\boldsymbol{B}}$ are the fixed control points, typically the boundary control points, since these will be treated differently. We let $n^{\boldsymbol{I}}$ and $n^{B}$ denote the number of inner and boundary control points, respectively. Note that sometimes some of the inner control points will be fixed, and some of the boundary control points might be free. Note that the same control points can have one component fixed and another component free. An example of such a scenario is seen later in Section 3.3, where the control points at a symmetry boundary will be allowed to move in the x-direction and not in the $y$-direction. When considering multipatch domains, we will often treat the control points on the patch interfaces as inner control points, such that they are determined by the parametrization technique.

So the challenge is how to determine the inner control points from the boundary control points, as illustrated in Figure 2.1.

When performing shape optimization with IGA, parameters that define the shape, for example the position of boundary control points, is iteratively updated during an optimization process. This makes the parametrization challenge even more important, as we need to maintain the geometry parametrization during the optimization process.


Figure 2.1: The parametrization challenge of finding a parametrization of the interior from the boundary curves.

Effectively this can limit the designs that you are able to consider, since they need to be parametrizable with the chosen strategy.

The parametrization can affect the accuracy of the approximation to a PDE [20], so it is important that it is of good quality. At the very least it should be a bijective map, meaning that the Jacobian determinant $\operatorname{det} J$ should be nonzero $\operatorname{det} J>0$.

We will in this chapter introduce some techniques for finding 2D parametrizations as benchmarks for the rest of the work. We will compare the techniques on some examples in the end of the chapter in Section 2.4.

### 2.1 Linear Parametrization Techniques

### 2.1.1 Coons patch

One of the most simple parametrization method is the Coons patch method. The idea here is to use a combination of linear interpolations to find the inner control points. We will here consider the 2D case, however the method can easily be extended to 3D.

Given four boundary curves $\gamma_{i}:[0,1] \rightarrow \mathbb{R}^{2}$ for $i=1, \ldots, 4$ that meets in the four corners

$$
\begin{align*}
& \gamma_{1}(0)=\gamma_{3}(0),  \tag{2.2a}\\
& \gamma_{1}(1)=\gamma_{4}(0),  \tag{2.2b}\\
& \gamma_{2}(0)=\gamma_{3}(1),  \tag{2.2c}\\
& \gamma_{2}(1)=\gamma_{4}(1), \tag{2.2~d}
\end{align*}
$$

as illustrated in Figure 2.1, we can interpolate the opposing sides linearly as

$$
\begin{equation*}
L_{1}(s, t)=(1-t) \gamma_{1}(s)+t \gamma_{2}(s) \tag{2.3}
\end{equation*}
$$

and

$$
\begin{equation*}
L_{2}(s, t)=(1-s) \gamma_{3}(t)+s \gamma_{4}(t) . \tag{2.4}
\end{equation*}
$$

These linear interpolations are the first two terms that are illustrated in Figure 2.2. Now if we add these two and subtract the bilinear interpolation of the four corners

$$
\begin{equation*}
B(s, t)=(1-s)(1-t) \gamma_{1}(0)+s(1-t) \gamma_{1}(1)+(1-s) t \gamma_{2}(0)+s t \gamma_{2}(1) . \tag{2.5}
\end{equation*}
$$

as

$$
\begin{equation*}
G(s, t)=L_{1}(s, t)+L_{2}(s, t)-B(s, t) \tag{2.6}
\end{equation*}
$$

then $G$ will be a map such that

$$
\begin{equation*}
G \upharpoonright_{t=0}=\gamma_{1}, G \upharpoonright_{t=1}=\gamma_{2}, G \upharpoonright_{s=0}=\gamma_{3}, G \upharpoonright_{s=1}=\gamma_{4} . \tag{2.7}
\end{equation*}
$$

If the boundary curves are given as splines such that $\gamma_{1}$ and $\gamma_{2}$ shares one spline space and $\gamma_{3}$ and $\gamma_{4}$ share another spline space, then this operation can be performed directly on the control points, as illustrated in Figure 2.2.


Figure 2.2: Illustration of the Coons patch parametrization method. The blue dots are control points, and the dashed lines are interpolations.

The Coons patch method is a linear parameterization method, in that the position of the inner control points will depend linearly on the boundary control points, but it will not always provide a valid parameterization [43].

### 2.1.2 Spring method

Another linear parameterization methods is the so called spring method. With this method the inner control points are found such that every inner control points are


Figure 2.3: With the spring method we imagine springs between all the control points, then we choose the positions of the inner control points such that this system is balanced.
an average of its neighbours. It corresponds to attaching springs with equal spring constants between neighboring control points and 'releasing' the system, when keeping the boundary control points fixed. This is sketched in Figure 2.3. This corresponds to solving the linear system

$$
\begin{equation*}
K \boldsymbol{c}=0 \tag{2.8}
\end{equation*}
$$

where the entries in the matrix $K$ is given as

$$
K_{i j}= \begin{cases}1 & \text { if } c_{i} \text { and } c_{j} \text { are neighbours }  \tag{2.9}\\ -4 & \text { if } i=j \\ 0 & \text { otherwise }\end{cases}
$$

Note that since we fix the boundary and fixed control points these can be moved to the right hand side in equation (2.8).

The spring method can sometimes be used when the Coons patch method fail to produce valid parametrizations, however it still fails to generate valid parametrizations if the shape is too complicated as we shall see in Section 2.4.

### 2.2 Validity Constraints

In this section we will introduce constraints to ensure that the geometry parametrization is valid, i.e., $\operatorname{det} J \neq 0$. Without loss of generality we will consider the case $\operatorname{det} J>0$.

In the literature there are a few different methods for introducing constraints that ensure $\operatorname{det} J>0$. One method is using injectivity cones [45]. In this work we will use the more common method, where you expand the Jacobian determinant and introduce constraints on the expansion coefficients [20].

The idea is that if we consider a geometry map of degrees $\boldsymbol{p}$ with knotvectors $\boldsymbol{\Xi}$ such that $G \in\left(\mathcal{S}_{\boldsymbol{\Xi}}^{\boldsymbol{p}}\right)^{d}$ with $G \in C^{k}$, then the Jacobian determinant $\operatorname{det} J \in C^{k-1}$ is also a spline

$$
\operatorname{det} J \in \mathcal{S}_{\underline{\widehat{\underline{E}}}}^{\widehat{\hat{p}}}
$$

with $\widehat{\boldsymbol{p}}=d \cdot \boldsymbol{p}-1$ and where $\widehat{\boldsymbol{\Xi}}_{i}$ contains the same knots as $\boldsymbol{\Xi}_{i}$ but with the multiplicity of the knots increase by $(d-1) p$ to account for the increase in degree and decrease in differentiability. We will denote the space $\mathcal{S}_{\widehat{\underline{p}}}^{\widehat{\boldsymbol{p}}}$ by $\mathcal{S}_{\text {det }}$ to simplify the notation.

This means that we can expand $\operatorname{det} J$ in this space as

$$
\begin{equation*}
\operatorname{det} J=\sum_{i=1}^{\widetilde{n}} d_{i} \widehat{\mathcal{R}}_{i}(\boldsymbol{\xi}) \tag{2.10}
\end{equation*}
$$

We can now introduce the constraints

$$
\begin{equation*}
\boldsymbol{d} \geq \varepsilon, \tag{2.11}
\end{equation*}
$$

where $\boldsymbol{d}=\left(d_{1}, \ldots, d_{\widetilde{n}}\right)$, and $\varepsilon>0$. Since the tensor product B-splines $\widehat{R}_{i}$ are nonnegative, the constraints (2.11) are sufficient, meaning that if $\boldsymbol{d} \geq \epsilon$ holds then also $\operatorname{det} J>0$. However it is not a necessary condition, since we can have $\operatorname{det} J>0$ with $\boldsymbol{d} \ngtr 0$.

### 2.2.1 Relaxing the constraints

That the condition is only necessary means that it is sometimes too strict. One way to relax the constraints is to expand the determinant in a larger spline space. Such a spline space can be obtained by refining $\mathcal{S}_{\text {det }}$. It is well known that when a spline is uniformly refined the control polygon, i.e., the expansion coefficients, move closer to the value of the spline, in this case the Jacobian determinant, for a proof and further details we refer the reader to $[13,9]$. We will denote by $\mathcal{S}_{\text {det }}^{(k)}$ the spline space obtained by applying uniform refinement k-times to $\mathcal{S}_{\text {det }}$. We will denote the spline coefficients on the kth refinement level as $\boldsymbol{d}^{(k)}$. Note that we can use local refinement instead of uniform refinement, as long as the basis satisfy the partition of unity property and is non negative. We shall later in section 3.1 use by using Truncated Hierarchical B-splines [19]. This can be used to relax the constraints (2.11) while keeping the number of constraints low. This strategy will be considered later in the context of shape optimization in Section 3.1.

### 2.2.2 Computing expansion coefficients

In this work we will compute the expansion coefficients $\boldsymbol{d}$ by projection. However they can also be computed by interpolation. In both methods we end up solving a linear system of equations to obtain $\boldsymbol{d}$. When using projection the coefficients can be found by solving the linear system

$$
\begin{equation*}
M \boldsymbol{d}=\boldsymbol{D} \tag{2.12}
\end{equation*}
$$

where the entries in $M$ and $\boldsymbol{d}$ are given by

$$
\begin{align*}
M_{i j} & =\int_{[0,1]^{d}} \widehat{\mathcal{R}}_{i} \widehat{\mathcal{R}}_{j} \mathrm{~d} \boldsymbol{\xi}  \tag{2.13a}\\
D_{j} & =\int_{[0,1]^{d}} \operatorname{det} J \widehat{\mathcal{R}}_{j} \mathrm{~d} \boldsymbol{\xi} \tag{2.13b}
\end{align*}
$$

### 2.2.3 Note on these as constraints

When using the constraints (2.11) as constraints for an optimization problem it is possible to avoid having to invert the mass matrix $M .{ }^{1}$ This can be done by introducing slack variables to rewrite the constraint

$$
\boldsymbol{d}=M^{-1} \boldsymbol{D} \geq \varepsilon
$$

as

$$
\begin{align*}
& s=M^{-1} \boldsymbol{D}  \tag{2.14a}\\
& s \geq \varepsilon \tag{2.14b}
\end{align*}
$$

Now we can multiply with $M$ on both sides of (2.14a) to get

$$
\begin{align*}
M s & =D  \tag{2.15a}\\
s & \geq \varepsilon \tag{2.15b}
\end{align*}
$$

Note that even though these might look like linear constraints, one has to remember that det $J$ and therefore $\boldsymbol{D}$ depend non-linearly on the control points. This approach will not be used in this work.

### 2.2.4 Number of constraints

In Table 2.1 we show the number of constraints required for different combinations of number of interior knots and degrees, for 2D and 3D tensor product splines. Note that in practice the constraints might be too strict and to relax the constraint the space $\mathcal{S}_{\text {det }}$ has to be refined which increases the number of constraints.

[^2]|  | 2 D |  |  | 3 D |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Interior knots | $p=1$ | $p=2$ | $p=3$ | $p=1$ | $p=2$ | $p=3$ |
| 2 | 36 | 100 | 196 | 729 | 4096 | 12167 |
| 4 | 100 | 256 | 484 | 3375 | 17576 | 50653 |
| 8 | 324 | 784 | 1444 | 19683 | 97336 | 274625 |
| 16 | 1156 | 2704 | 4900 | 132651 | 636056 | 1771516 |

Table 2.1: The number of spline coefficients of the Jacobian determinant.

### 2.2.5 Maximization of the determinant

One way to use these constraints to generate a valid parametrization is by maximizing the smallest spline coefficient of the Jacobian determinant. This does not necessarily give a parametrization of good quality, but it provide us with a way of generating a valid initial guess for other methods. The following minimization problem is considered

$$
\begin{equation*}
\max _{c^{I}} \min _{i} d_{i} \tag{2.16}
\end{equation*}
$$

which can be rewritten by introducing a slack variable $s$ :

$$
\begin{aligned}
& \min _{\boldsymbol{c}^{I}, z}-s \\
& \text { s.t. } \boldsymbol{d} \geq s
\end{aligned}
$$

### 2.3 Optimization Based Methods

In this section we will consider a class of parameterization techniques based on nonlinear optimization. The idea is that if we introduce a local quality metric $q(\boldsymbol{c}, \boldsymbol{\xi})$ that is small for parametrizations of good quality and large for parameterizations of poor quality, then we can try to minimize the global quality metric,

$$
\begin{equation*}
Q(\boldsymbol{c})=\int_{[0,1]^{d}} q(\boldsymbol{c}, \boldsymbol{\xi}) \mathrm{d} \boldsymbol{\xi} \tag{2.17}
\end{equation*}
$$

to achieve a parametrization of high quality. If we minimize over the inner control points we arrive at the following minimization problem

$$
\begin{equation*}
\min _{\boldsymbol{c}^{I}} Q(\boldsymbol{c}) \tag{2.18}
\end{equation*}
$$

In some cases we will add the constraints on the Jacobian determinant in (2.11) as

$$
\begin{align*}
& \min _{\boldsymbol{c}^{I}} Q(\boldsymbol{c})  \tag{2.19a}\\
& \text { s.t. } \boldsymbol{d}>\varepsilon \tag{2.19b}
\end{align*}
$$

But in order to use many of these methods we need a valid initial guess. We will first cover one way to get a valid initial guess, and the move on to introduce a number of quality metrics. We will now move on to consider some different quality metrics

### 2.3.1 Liao functional

One choice of $m$ is the Liao functional [27]. If we let $g$ be the first fundamental form

$$
g=J^{T} J=\left[\begin{array}{ll}
g_{11} & g_{12}  \tag{2.20}\\
g_{21} & g_{22}
\end{array}\right]=\left[\begin{array}{cc}
x_{\xi}^{2}+y_{\xi}^{2} & x_{\xi} x_{\eta}+y_{\xi} y_{\eta} \\
x_{\xi} x_{\eta}+y_{\xi} y_{\eta} & x_{\eta}^{2}+y_{\eta}^{2}
\end{array}\right]
$$

then the Liao measure $m_{L}$ is given by the Frobienius norm of $g$

$$
\begin{equation*}
q_{L}=g_{11}^{2}+2 g_{12}^{2}+g_{22}^{2} \tag{2.21}
\end{equation*}
$$

When using the Liao functional will will consider the minimization problem (2.19), i.e., we will use the constraint on the Jacobian determinant, in order for the method to produce valid parametrizations. One issue we faced when trying to use the Liao functional for shape optimization is that after finding a minimizer to (2.19) there are active constraints, i.e., there exists $i$ such that $d_{i}=\varepsilon$. This is a disadvantage due to two reasons. Firstly it might produce parametrizations which are close to being degenerate and can lead to discretization error when using IGA. Secondly when using an optimization based parametrization strategy in a shape optimization algorithm, one can linearize the method by minimizing a quadratic approximation to the quality metric, as shall be considered in Chapter 3. However this only works well if we linearize at a reference parametrization where the metric $Q_{L}$ has a local minimum, which is not the case when there are active constraints. This issue arise since the method sometimes would produce non valid parametrizations if the constraints 2.11. In [27] they suggest to combat this by using the modified Liao functional.

The modified Liao functional is given as

$$
\begin{equation*}
m_{M L}=\left(\frac{g_{11}+g_{22}}{\operatorname{det} J}\right)^{2}=\left(\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J}\right)^{2} . \tag{2.22}
\end{equation*}
$$

Here the we divide by $\operatorname{det} J$ which drives the optimization away from maps with small Jacobian determinant.

### 2.3.2 Winslow functional

The Winslow functional is given by the quality measure

$$
\begin{equation*}
m_{W}=\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J^{2 / d}} \tag{2.23}
\end{equation*}
$$

It was introduced in the context of grid generation in finite elements [42]. It was introduced for the 2D case $d=2$, and there are different methods to generalize it to

3D. In this work we shall use the generalization (2.23). The factor $2 / d$ is chosen such that the measure is scaling invariant. We use the notation

$$
\begin{equation*}
W(G)=\int_{[0,1]^{d}} \frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J^{2 / d}} \mathrm{~d} \boldsymbol{\xi} \tag{2.24}
\end{equation*}
$$

For $d=2$ it is the square root of the modified Liao functional. It has some nice mathematical properties in 2D. First of all it serves as a measure of conformality. We let $\lambda_{1}$ and $\lambda_{2}$ denote the eigenvalues of the first fundamental form $g=J^{T} J$. Then the map is conformal if $\lambda_{1}=\lambda_{2}$. If we assume that $\lambda_{1}>0$ and $\lambda_{2}>0$ we have

$$
\begin{aligned}
\frac{\left(\sqrt{\lambda_{1}}-\sqrt{\lambda_{2}}\right)^{2}}{\sqrt{\lambda_{1} \lambda_{2}}} & =\frac{\lambda_{1}+\lambda_{2}-2 \sqrt{\lambda_{1} \lambda_{2}}}{\sqrt{\lambda_{1} \lambda_{2}}} \\
& =\frac{\lambda_{1}+\lambda_{2}}{\sqrt{\lambda_{1} \lambda_{2}}}-2
\end{aligned}
$$

The above equation is always larger than zero, and since it attains the value zero if the map is conformal, it is clear that it is minimal for conformal maps. The Winslow functional is given by the term $\frac{\lambda_{1}+\lambda_{2}}{\sqrt{\lambda_{1} \lambda_{2}}}$ since

$$
\begin{equation*}
\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J}=\frac{\operatorname{tr}(g)}{\sqrt{\operatorname{det} g}}=\frac{\lambda_{1}+\lambda_{2}}{\sqrt{\lambda_{1} \lambda_{2}}} \tag{2.25}
\end{equation*}
$$

So when we minimize the Winslow functional we look for a map that is as conformal as possible. Another nice property of the Winslow functional is that is has a unique minimum and the inverse of its minimizer is a pair of harmonic functions that is a diffeomorphism on the interior [42, 20]. This result, to the best of my knowledge, only holds in 2D. In spite of this result the authors in [20] propose to use constraints on the Jacobian determinant to safeguard against numerical errors. In our experience this can be avoided if we set $W=\infty$ if $\operatorname{det} J \leq 0$ at one of the quadrature points used for the integration. This will ensure that the optimization does not find a parametrization with $\operatorname{det} J \leq 0$ in one of the quadrature points. However note that with this modification $m_{W}$ is no longer continuous. For example consider the scaling $G(\boldsymbol{\xi})=a \cdot \boldsymbol{\xi}$, for $a \in \mathbb{R}$. When $a \rightarrow 0$ then $\operatorname{det} J \rightarrow 0$ but $m_{W}=2$. But with this modification the step in the line search part of an optimization algorithm is only accepted when $\operatorname{det} J>0$ at the quadrature points. However it requires that we start the optimization from a valid parametrization.

### 2.3.3 Harmonic functional

The final metric that we will cover in this section will be referred to as the 'Harmonic functional' in lack of a better name. It is proposed in [43]. The idea is, similar to the Winslow functional, to look for a map whose inverse is a pair of harmonic functions.

To accomplish this the authors introduce the measure $(L x)^{2}+(L y)^{2}$ where $L$ is given as

$$
\begin{equation*}
L=\left(x_{\eta}^{2}+y_{\eta}^{2}\right) \frac{\partial^{2}}{\partial \xi^{2}}-2\left(x_{\xi} x_{\eta}+y_{\xi} y_{\eta}\right) \frac{\partial^{2}}{\partial \xi \eta}+\left(x_{\xi}^{2}+y_{\xi}^{2}\right) \frac{\partial^{2}}{\partial \eta^{2}} \tag{2.26}
\end{equation*}
$$

where $G=(x, y)$. It can be written in closed form as

$$
\begin{equation*}
L x=\operatorname{adj}\left(J^{T} J\right): H(x) \tag{2.27}
\end{equation*}
$$

and similarly for $L y$. Here : denotes the Frobenius inner product and $H(x)$ denotes the Hessian of the map $x$, and $\operatorname{adj}(\cdot)$ is the adjugate matrix. The authors then further add the two terms to achieve a parametrization with good orthogonality arriving at

$$
\begin{equation*}
m_{H}=(L x)^{2}+(L y)^{2}+\epsilon_{1}\|H(x)\|_{F}^{2}+\epsilon_{2}\|J\|_{F}^{2} \tag{2.28}
\end{equation*}
$$

In [43] the authors demonstrate that this method can produce valid parametrizations without the use of constraints on the Jacobian determinant. We were however not able to reproduce these results.

### 2.4 Examples

In this section we will consider the previously proposed parametrization methods on three different examples. All of the example are in 2D. Examples in 3D will be considered in Chapter 4.

In Figure 2.4 we will consider a domain that we will call the 'starfish', which resembles the domain in Figure 4 in [43]. Additionally we will consider two jigsaw puzzle pieces similar to the ones from [20], the one in Figure 2.5 we will refer to as jigsaw 2 and the one in Figure 2.6 we will refer to as jigsaw 1 . The methods is implemented in G+Smo and we use the Ipopt optimization library for the optimization.

We will for each example report the smallest spline coefficients of the Jacobian determinant $\min \boldsymbol{d}$ if the parametrization is valid, i.e., if $\operatorname{det} J>0$. If it was necessary to refine the spline space $\mathcal{S}_{\text {det }} \mathrm{k}$ times to obtain only positive coefficients we will denote the smallest one as $\min \boldsymbol{d}^{(k)}$. We use $\varepsilon=10^{-5}$ as a lower bound for the Jacobian determinant constraints.

In Figure 2.4 we see the results for the starfish. Notably we see that the spring method and the Harmonic functional without constraints fails to produce a valid parametrization. Furthermore we see that many of the optimization based methods has active constraints, since $\min \boldsymbol{d} \approx \varepsilon=10^{-5}$. When using the Winslow functional without constraints we arrive at a valid parametrization, but it was necessary to refine the $\mathcal{S}_{\text {det }}$ thrice to get all positive coefficients.

In Figure 2.5 we consider jigsaw 2, which is the more simple of the two jigsaw puzzle pieces. The results here are similar as for the starfish domain.

In Figure 2.6 we consider jigsaw 1, but here even the method where we maximize the smallest spline coefficient fails to find a valid parametrization. This can mean
three things, either there is no valid spline parametrization on the given refinement level, the constraints on the Jacobian determinant are too strict or it could be that the optimization get stuck in a local minimum. We will later see that it is actually possible to find a valid parametrization on this refinement level, when we in Chapter 4 develop a new approach to finding parametrizations. With this approach we are able to find a valid parametrization for this domain. However to prove that it is valid we need to refine $\mathcal{S}_{\text {det }}$ several times, which indicates that the constraints considered here might be too strict, for this domain. As we need a valid parametrization as an initial guess for the other methods (expect for the harmonic functional) we cannot use these parametrization methods.

In Figure 2.7 we consider jigsaw 1 but using uniformly refined splines, which allows us to find valid parametrizations. Note the reason the we are able to find a valid parametrizations might not only be due to the fact that we have more degrees of freedom to work with, but also due to the fact that the constraints on the Jacobian determinant will also be less strict for this refinement. We observe that in this case the parametrization found by minimizing the Winslow functional has no active constraints.


Figure 2.4: Parameterization for the starfish

(a) Spring, $\operatorname{det} J \ngtr 0$

(c) Liao, $\min \boldsymbol{d}=9.99 \cdot 10^{-5}$

(e) Winslow with constraints, $\min \boldsymbol{d}=3.18 \cdot 10^{-5}$

(g) Winslow without constraints, $\min \boldsymbol{d}^{(1)}=9.47$

(b) $\max \min d_{i}, \min \boldsymbol{d}=47.4$

(d) Modified Liao, $\min \boldsymbol{d}=1.69$. $10^{-5}$

(f) Harmonic with constraints, $\min \boldsymbol{d}=9.99 \cdot 10^{-5}$

(h) Harmonic without constraints, $\operatorname{det} J \ngtr 0$

Figure 2.5: Parameterization for jigsaw 2


Figure 2.6: Parameterization for jigsaw 1


Figure 2.7: Parameterization for jigsaw 1

## снартв 3

## 2D Shape

## Optimization using IGA

In this chapter we will discuss different approaches for doing shape optimization with Isogeometric Analysis. Some of the figures in this chapter is taken directly from the manuscript in Appendix A. The problem we are interested in are on the following form

$$
\begin{align*}
& \max _{\Omega \in \mathcal{O}_{a d}} E(\Omega, \tilde{u}),  \tag{3.1a}\\
& \text { s.t. } \tilde{a}_{\Omega}(\tilde{u}, \tilde{v})=\tilde{\ell}_{\Omega}(\tilde{v}) \quad \text { for all } \tilde{v} \in \tilde{V}, \tag{3.1b}
\end{align*}
$$

where $E$ is the objective and $\mathcal{O}_{a d}$ is the set of admissible shapes. The constraint (3.1b) is the weak formulation of the PDE that governs the physics. This could for example be the ones considered in Section 1.2. When using Isogeometric Analysis the solution this weak formulation is approximated by a spline, which coefficients bu can be found by solving a linear system

$$
\begin{equation*}
A_{c} \boldsymbol{u}=\boldsymbol{F}_{\boldsymbol{c}} \tag{3.2}
\end{equation*}
$$

Where $A_{\boldsymbol{c}}$ and $\boldsymbol{F}_{\boldsymbol{c}}$ are the system matrix and right hand side that defines the linear system which arise from the discretization. The subscript $c$ is to indicate that this matrix and vector depends on the control points $\boldsymbol{c}$. By introduction a spline parametrization for representing the physical domain, we can write a discrete shape optimization problem as

$$
\begin{align*}
\max _{\boldsymbol{\alpha}} & E(\boldsymbol{c}, \boldsymbol{u}),  \tag{3.3a}\\
\text { s.t. } & A_{\boldsymbol{c}} \boldsymbol{u}=f_{\boldsymbol{c}},  \tag{3.3b}\\
& \boldsymbol{c}=\boldsymbol{c}(\boldsymbol{\alpha}),  \tag{3.3c}\\
& \operatorname{det} J>0,  \tag{3.3d}\\
& \boldsymbol{\alpha}_{L} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{U}, \tag{3.3e}
\end{align*}
$$

where $\boldsymbol{\alpha}$ is the design variables. The vectors $\boldsymbol{\alpha}_{U}$ and $\boldsymbol{\alpha}_{L}$ are upper and lower bounds for the design variables. The constraint (3.3d) is to make sure that the parametrization is valid. In the shape optimization frameworks discussed in this work this constraint will be handled in two different ways. Firstly, in section 3.1 we will replace it
with constraints on the spline coefficients of the Jacobian determinant as described in section 2.2. Secondly, in section 3.2 the constraint (3.3d) will only be imposed implicitly by adding a regularization term to the objective, that tends to $\infty$ as det $J$ goes to 0 . Finally, the function $\boldsymbol{c}(\boldsymbol{\alpha})$ describes how the position of the control points depends on the design variables $\boldsymbol{\alpha}$. In this work we shall consider two different approaches. One where the design variables are the boundary control points and $\boldsymbol{c}(\boldsymbol{\alpha})$ is given by a linearization of the Winslow minimization problem. And one where all the control points will enter the formulation as design variables $\boldsymbol{c}(\boldsymbol{\alpha})=\boldsymbol{\alpha}$. . Another example will be where we optimize over all the control points, in which case $\boldsymbol{c}(\boldsymbol{\alpha})=\boldsymbol{\alpha}$.

We will in the following use the notation

$$
c=\left[\begin{array}{l}
c^{I}  \tag{3.4}\\
c^{B}
\end{array}\right]
$$

where $\boldsymbol{c}^{\boldsymbol{I}}$ are the inner control points and $\boldsymbol{c}^{\boldsymbol{B}}$ are the boundary control points. We let $n^{\boldsymbol{I}}$ and $n^{\boldsymbol{B}}$ denote the number of inner and boundary control points, respectively.

### 3.1 Shape Optimization using Linearizations

In this chapter we will consider a shape optimization framework that is based on using linearizations of the minimization of the Winslow functional, described in Section 2.

In the following the position of the boundary control points will be the design variables for the shape optimization $\boldsymbol{\alpha}=\boldsymbol{c}^{\boldsymbol{B}}$, while the position of the inner control points is determined by the parametrization strategy, as a function of $\boldsymbol{c}^{I}$. With this distinction the boundary control points are not limited to the control points of the boundary of the physical domain, it could also be control points that represent an interface between two patches in a multipatch domain, as it will be the case in the model problem considered in Section 3.3

The goal is to use the parametrization strategy based on minimization of the Winslow functional for shape optimization. With this strategy the position of the inner control points of the spline parametrization is found as a minimizer to the following minimization problem

$$
\begin{equation*}
\min _{\boldsymbol{c}^{I}} W(\boldsymbol{c}) \tag{3.5}
\end{equation*}
$$

where

$$
\begin{equation*}
W=\int_{\hat{\Omega}} \frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J} \mathrm{~d} \boldsymbol{\xi} \tag{3.6}
\end{equation*}
$$

with $J$ the Jacobian of the geometry map. Here the design variables are the inner control points $\boldsymbol{c}^{\boldsymbol{I}}$ while the boundary control points $\boldsymbol{c}^{\boldsymbol{B}}$ are fixed. During a shape optimization process, the shape of the domain is updated iteratively, and it is therefore necessary to update the position of the inner control points for each iteration of the shape optimization problem. To avoid having to solve the minimization problem (3.5) repeatedly, we linearize this minimization problem, following to the method in [33].

Given a reference parametrization $\boldsymbol{c}_{0}$ the problem (3.5) can be linearized by replacing $W$ with a second order Taylor expansion, giving the following optimization problem

$$
\begin{equation*}
\min _{\Delta \boldsymbol{c}^{I}} \frac{1}{2} \Delta \boldsymbol{c}^{T} H\left(\boldsymbol{c}_{0}\right) \Delta \boldsymbol{c}+\nabla W\left(\boldsymbol{c}_{0}\right)^{T} \Delta \boldsymbol{c}+W\left(\boldsymbol{c}_{0}\right) \tag{3.7}
\end{equation*}
$$

Where $\Delta \boldsymbol{c}=\boldsymbol{c}_{0}+\boldsymbol{c}$. The minimizer to this problem can be found by solving the linear system

$$
\begin{equation*}
{\frac{\partial \boldsymbol{c}^{\boldsymbol{c}}}{}}^{T} H\left(\boldsymbol{c}_{0}\right) \Delta \boldsymbol{c}=-{\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{c}^{\boldsymbol{I}}}}^{T} \nabla W\left(\boldsymbol{c}_{0}\right) \tag{3.8}
\end{equation*}
$$

We can eliminate the boundary and fixed control points since both are fixed, yielding the system

$$
\begin{equation*}
H_{c^{I}, c^{I}} \Delta \boldsymbol{c}^{I}=-\nabla W_{c^{I}}-H_{c^{I}, c^{B}} \Delta c^{B} \tag{3.9}
\end{equation*}
$$

where $\left[H_{c^{I}, c^{I}}\right]_{i j}=\partial^{2} W / \partial c_{i}^{I} \partial c_{j}^{I},\left[H_{c^{I}, c^{B}}\right]_{i j}=\partial^{2} W / \partial c_{i}^{I} \partial c_{j}^{B}$ and $\left[\nabla W_{c^{I}}\right]_{i}=\partial W / \partial c_{i}^{I}$.
Now if we let $\Delta \boldsymbol{c}^{\boldsymbol{I}}$ be given by the linear system (3.9) we can define the parametrization strategy as follows. Given new positions of the boundary control points we can find the position of all the control points as

$$
c=c_{0}+\Delta c
$$

where $\Delta c=\left[\begin{array}{ll}\Delta c^{\boldsymbol{I}} & \Delta \boldsymbol{c}^{\boldsymbol{B}}\end{array}\right]^{T}$
Until now we have not discussed how to choose the reference parametrization $\boldsymbol{c}_{0}$. In principle it could be chosen arbitrarily. However the second order expansion of $W$ minimized in (3.7) is a good approximation only in a neighborhood of $\boldsymbol{c}_{0}$. So we want to choose $\boldsymbol{c}_{0}$ such that the result of the parametrization strategy given by (3.7) stays near $\boldsymbol{c}_{0}$. One way to guarantee this is to require that $\boldsymbol{c}\left(\boldsymbol{c}_{k}^{B}\right)=\boldsymbol{c}_{k}$, which means that if we don't change the boundary control points, ie. $\Delta \boldsymbol{c}^{B}=0$, then also the solution to the linear system (3.9) is zero $\Delta \boldsymbol{c}^{\boldsymbol{I}}=0$. Inserting $\Delta \boldsymbol{c}^{\boldsymbol{B}}=0$ into (3.9) we get

$$
H_{c^{I}, c^{I}} \Delta \boldsymbol{c}^{I}=-\nabla W_{c^{I}}
$$

which means that $\nabla W_{c^{I}}=0$ if $\Delta \boldsymbol{c}^{I}=0$. In other words the reference parametrization in that case has to be a minimizer of the original problem (3.5).

We can now use this parametrization strategy to define the subproblem

$$
\begin{array}{ll}
\max _{\boldsymbol{\alpha}} & E(\boldsymbol{c}, \boldsymbol{u}), \\
\text { s.t. } & K_{\boldsymbol{c}} \boldsymbol{u}=f_{\boldsymbol{c}}, \\
\boldsymbol{c}=\boldsymbol{c}_{0}+\Delta \boldsymbol{c}, & \text { The discretized PDE } \\
\boldsymbol{c}^{\boldsymbol{B}}=\boldsymbol{\alpha}, & \text { The parametrization strategy } \\
\boldsymbol{d} \geq \varepsilon, & \text { How } \boldsymbol{c} \text { depends on } \boldsymbol{\alpha} \\
\boldsymbol{\alpha}_{L} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{U} . & \text { Positivity of det } J  \tag{3.10f}\\
\text { Design bounds }
\end{array}
$$

Note here that the only difference from (3.3) is (3.10c) and (3.10d) which defines the parametrization strategy and the fact that we optimize over the positions of a subset
of the control points, that defines the linearization based parametrization strategy. We can now solve a sequence of these problems where there reference $\boldsymbol{c}_{0}$ is updated between in each problem by solving (3.5) to improve the parametrization.

### 3.1.1 Strategy for Validity Constraints

In [33] the constraints described in Section 2.2 have been added to the minimization of the Winslow functional in (3.5) arriving at the problem

$$
\begin{align*}
& \min _{\boldsymbol{c}^{I}} W(\boldsymbol{c}),  \tag{3.11a}\\
& \text { s.t. } \boldsymbol{d} \geq \varepsilon . \tag{3.11b}
\end{align*}
$$

The rationale behind this is that a feasible point to the subproblem (3.10) needs to satisfy the constraint (3.10e). And since we linearize the parametrization strategy around $\boldsymbol{c}_{0}$ it is natural to require $\boldsymbol{c}_{0}$ to be feasible. This can be enforced by using the constraint (3.11b) when minimizing $W$. However the challenge with this approach is that if any of the constraints (3.11b) are active, meaning that there exists $i$ such that $d_{i}=\varepsilon$, then the problem (3.9) is not a good substitute for (3.11).

This can be seen by considering the case where $\Delta \boldsymbol{c}^{\boldsymbol{B}}=0$, meaning that the design variables are not changed. Since we have active constraints we might have $\nabla W_{\boldsymbol{c}^{I}} \neq 0$, which means that also $\Delta \boldsymbol{c}^{I} \neq 0$, i.e., the inner control points will change. This is an issue since it means that the inner control points will not stay in an neighborhood around $\boldsymbol{c}_{0}$, and we cannot guarantee that the parametrization will remain valid. Moreover the approximation (3.9) might not be a good approximation of $\left\|\Delta \boldsymbol{c}^{\boldsymbol{B}}\right\|$ is large. The way the authors of [33] deal with this issue is that if any of the constraints (3.11b) are active after solving (3.11) then they refine the geometry via knot insertion.

In this work we take a different approach. In our experiments we found that when solving the unconstrained problem (3.5) the minimizer would in fact have a positive determinant everywhere, however the condition $\boldsymbol{d}>0$ is not satisfied, since this condition is often to strict. In order to prove that $\operatorname{det} J>0$ we often need to relax the constraint, by refining the spline space in which the spline coefficients $\boldsymbol{d}$ is found. To avoid an excessive increase in the number of constraints we do this locally, using locally refinable truncated hierarchical B-Splines [19]. The refinement strategy goes as follows: After solving (3.5) we compute the spline coefficients of the Jacobian determinant $\boldsymbol{d}$. If $d_{i}<0$ for some $i$ then we refine the knot intervals where the basis function $\widehat{\mathcal{R}}_{i}$ has support. This is repeated until all $d_{i}$ is positive. The process is illustrated in Figure 3.1, where we see that even though the Jacobian determinant has negative spline coefficients at refinement level 0 , but not after refining locally twice.

This loop will only end if $\operatorname{det} J>0$, which we cannot guarantee. ${ }^{1}$ We can only guarantee that det $J$ is positive on all the gauss quadrature points used to calculate

[^3]$W$ in (3.5). For example it might not be possible to find a valid parametrization on the given discretization level. However in our experience this was not an issue when using the method on the 2D shape optimization that will be considered in Section 3.3. One explanation for this is that it is not an arbitrary shape that we try to parametrize using (3.5). It is the shape that is the solution to the previous subproblem (3.10), and therefore we know that there is a valid parametrization at the given discretization level, due to the constraint (3.10e). The advantages with this strategy is the number of control points used to represent the geometry is fixed, contrary to [33], and keep the number of constraints low during the optimization. This is further investigated on a specific shape optimization problem in Section 3.3.

The value $\varepsilon$ in the constraint (3.10e) will be chosen as $\varepsilon=\rho \min \boldsymbol{d}$, where $d_{i}$ for $i=1, \ldots, \widetilde{n}$ are the spline coefficients of the Jacobian determinant of the reference parametrization and for $0 \leq \rho \leq 1$. This is to ensure that the reference parametrization will be feasible. For the model problem considered in Section 3.3 we shall use $\rho=1 / 4$.

The full shape optimization algorithm is illustrated as a flowchart in Figure 3.2.

### 3.1.2 Implementation Details

In this work we use ipopt [41] for solving the subproblems (3.10). Ipopt is an Interior Point optimization algorithm implemented in C++.

The interior point algorithm requires some parameter tuning to work well, due to the fact that we start from the design found from the previous subproblem. Since ipopt uses an interior point algorithm, it pushes the starting guess away from the boundary to start at an interior point. The amount it is pushed can be adjusted with the parameter bound_push. It was our experience that using the standard value of 0.1 could lead to a violation of the constraints on the Jacobian determinant, since these constraint are quite sensitive to changes in the boundary control points. So to avoid this we needed to decrease the value of this parameter to $10^{-5}$.

Another parameter in ipopt is the barrier parameter $\mu$, which penalizes the constraints. We use the strategy monotone, where $\mu$ is decreased monotonically during the optimization. However we found that if the initial value of $\mu$, given by the parameter mu_init, were too large then it pushed the design towards designs with large Jacobian determinant. This is an issue since it means that the optimization is driven away from the starting guess $\boldsymbol{c}_{0}$ which is the point where the parametrization strategy is linearized. So to avoid this we found that we needed to decrease this parameter from the default value 0.1 to $10^{-4}$.

In general one can argue that an interior point algorithm might not be the best choice for solving problems that are 'warm started'. Here an Sequential Quadratic Program (SQP) might be more suited. The main reason that we use ipopt is that, besides being a state of the art optimization library, it is embedded in G+Smo as an extension, which makes the implementation easier. With the parameter tuning


Figure 3.1: An illustration of how the constraint $\boldsymbol{d}>\epsilon$ can be relaxed. In the top row we plotdet $J$ with its control net (spline coefficents) and the projection onto $\{z=0\}$, for different refinement levels. The negative spline coefficients are marked with a red circle. In the bottom row we plot the locally refined mesh. The mesh is refined at the support of the basis functions corresponding to negative coefficients.
explained above it seems to work well. For more details on the optimization algorithm used in ipopt see [41].


Figure 3.2: Flowchart of the optimization algorithm.

### 3.2 Shape Optimization using Regularization

In this section we will introduce a simpler framework for shape optimization with IGA, where the inner control points of the spline parametrization are included as design variables for the shape optimization. To ensure that the method produces valid parametrizations a regularization term $R_{\tau}$ is added to drive the optimization towards a parametrization of good quality. With this method a parametrization strategy is not needed, as the position of the inner control points is an outcome of the optimization. A similar method has been investigated in the context of shape optimization in mechanics in $[15,37]$.

In the following we will use the Winslow functional $W$ as the regularization term $\mathbb{R}_{\tau}=\tau W$, as it has proven to be a good quality measure for parametrizations. We will use numerical quadrature to evaluate $W$. It is important to note that we again set $W$ equal to $\infty$ if $\operatorname{det} J \leq 0$ in one of the quadrature points. When a line search method is used as part of the optimization process, then the step size will be chosen such that $W \neq \infty$, which means that $\operatorname{det} J>0$ is ensured at the quadrature points.

The optimization problem considered here is on the form

$$
\begin{array}{rl}
\min _{\boldsymbol{c}} & E(\boldsymbol{c}, \boldsymbol{u})+R_{\tau}, \\
\text { s.t. } & K_{\boldsymbol{c}} \boldsymbol{u}=f_{\boldsymbol{c}}, \\
& \boldsymbol{c}_{L} \leq \boldsymbol{c} \leq \boldsymbol{c}_{U} . \tag{3.12c}
\end{array}
$$

With this approach we avoid the explicit constraints on the coefficients of the Jacobian determinant, however we can only guarantee that it is positive in the quadrature points used when evaluating the Winslow functional. For our 2D model problem the designs found with this method still has positive Jacobian determinant everywhere, but we will see examples of the Jacobian determinant being negative between the quadrature points when we consider the method for 3D shape optimization, in chapter 4 and 5 . We found that the method was less prone to this when using GaussLobatto quadrature to evaluate the Winslow functional.

The downside with this method is that an appropriate value of $\tau$ needs to be chosen. If $\tau$ is too large the regularization term will dominate the objective and the final shape will be 'easy' to parametrization but might not have a low objective $E$. If $\tau$ is chosen too small the final parametrization might be of bad quality, which can lead to discretization errors in the analysis. This was also what we saw in experiments, as will be elaborated in Section 3.3. One method to chose an appropriate value of $\tau$ is to solve a sequence of the shape optimization problems (3.12) with decreasing $\tau$. One can for example evaluate the results by computing the objective using a finer discretization, to validate the performance of the shape and check if the optimization has exploited numerical errors.

### 3.2.1 Shape Optimization by Regularizations of Deformations

Sometimes it can be a disadvantage that the regularization term drives the optimization towards shapes that are easy to parametrize, for example if one knows that the optimal design is close to the initial design.

One way to modify the method is to regularize the deformation of the initial domain, instead of the geometry map $G$. We can split the geometry map $G$ into to parts $G=\widetilde{G} \circ G_{0}$. Here $G_{0}$ is a reference geometry map, for example the initial guess for a shape optimization problem. The setup is sketched in Figure 3.3.

We can compute the Winslow functional of the map $\widetilde{G}$ as

$$
\begin{equation*}
W(\widetilde{G})=\int_{\widetilde{\Omega}} \frac{\operatorname{tr}\left(\widetilde{J}^{T} \widetilde{J}\right)}{\operatorname{det} \widetilde{J}^{2}} \mathrm{~d} \tilde{x} \tag{3.13}
\end{equation*}
$$

Using the chain rule we get that

$$
\begin{equation*}
\widetilde{J} \circ G_{0}=J_{0}^{-1} J, \tag{3.14}
\end{equation*}
$$



Figure 3.3: We can use regularization of the deformation $\widetilde{G}$ to the initial domain.
which can be used to pull back the integral (3.13):

$$
\begin{equation*}
W(\widetilde{G})=\int_{[0,1]^{d}} \frac{\operatorname{tr}\left(J^{T} J_{0}^{-T} J_{0}^{-1} J\right)}{\operatorname{det} J^{\frac{2}{d}}} \operatorname{det} J_{0}^{\frac{2}{d}+1} \mathrm{~d} \tilde{x} \tag{3.15}
\end{equation*}
$$

Now using the simple modification of using $R_{\tau}=\tau W(\widetilde{G})$ means that we regularize the deformation of $G_{0}$ instead of $G$. I did not have time to investigate this approach further in this work.

### 3.3 Optimization of Electromagnetic Reflectors

We will in this section consider the problem of designing electromagnetic reflectors such that the electrical energy is maximized near a chosen point. It will serve as a model problem with which we will compare the two strategies presented in Chapter 3 for performing shape optimization with IGA. A similar problem has been considered in [33] and [1].

### 3.3.1 Problem description

We consider a 2D scattering problem, where a uniform electromagnetic wave travels in a dielectric medium (air) and is scattered by two opposing reflectors made from gold. We let $\epsilon_{c r}$ and $\mu_{c r}$ denote the electrical permittivity and permeability of a material, respectively. The wave travels in the x -direction so we will assume that the two reflectors are symmetric around the x -axis and only optimize over one of them. We use a first order absorbing boundary condition and consider a half circle as the computational (truncated) domain. The setup is illustrated in Figure 3.4. The electromagnetic field is found by solving the following PDE with the first order absorbing boundary condition at the boundary of the truncated domain.

$$
\begin{align*}
\nabla \cdot\left(\frac{1}{\epsilon_{c r}} \nabla \hat{u}\right)-k_{0}^{2} \mu_{r} \hat{u}=0 & & \text { in } \Omega,  \tag{3.16a}\\
\frac{\partial\left(\hat{u}-u^{i}\right)}{\partial n}+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right)\left(\hat{u}-u^{i}\right)=0 & & \text { on } \Gamma_{t} . \tag{3.16b}
\end{align*}
$$



Figure 3.4: Sketch of the shape optimization problem. The goal is to find a shape of the reflector that maximizes the field close to a point.
where $\hat{u}$ is the electric field, $k_{0}=2 \pi \sqrt{\epsilon_{0} \mu_{0}}$ is the wave number where $\epsilon_{0}$ and $\mu_{0}$ is the permittivity and permeability of free space, respectively. The radius of the truncated domain is given by $r_{t}$ and j denotes the imaginary unit. The far field $u^{i}$ is given as

$$
u^{i}(x, y)=\mathrm{e}^{-\mathrm{j} k_{0} \sqrt{\epsilon_{c r} \mu_{c r}} \cdot x} .
$$

As an objective we use the following energy function

$$
E(\Omega, \hat{u})=\int_{\Omega} \delta|\hat{u}|^{2} \mathrm{~d} x
$$

Where $\hat{u}$ is the solution to the $\operatorname{PDE}(3.16)$ and $\delta$ is given by a Gaussian function around $(0,0)$

$$
\delta(x, y)=\mathrm{e}^{\left(x^{2}+y^{2}\right) /\left(2 \alpha^{2}\right)}
$$

with $\alpha=0.1$. The physical parameters used for this problem is given in Table 3.1. The complex permitivitty of the reflector is calculated as $\epsilon_{c r}^{s}=\epsilon_{r, \text { gold }}-\mathrm{j} \frac{\sigma}{\omega \epsilon_{0}}$

### 3.3.2 Discretization

In this section we will briefly review the discretization of the problem above.
To discretize the PDE (3.16) we write it in the weak formulation: Find $u \in H^{1}(\Omega)$ such that the equation

$$
\begin{align*}
& \int_{[0,1]^{2}} \frac{1}{\epsilon_{c r}} \hat{\nabla} \hat{u} \cdot \hat{\nabla} \hat{v} \mathrm{~d} x+k_{0}^{2} \int_{[0,1]^{2}} \mu_{r} \hat{u} \hat{v} \mathrm{~d} x+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) \int_{\Gamma_{t}} \frac{1}{\epsilon_{c r}} \hat{u} \hat{v} \mathrm{~d} s \\
&=\frac{1}{\epsilon_{c r}} \int_{\Gamma_{t}}\left(\frac{\partial u^{i}}{\partial n}+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) u^{i}\right) \hat{v} \mathrm{~d} s \tag{3.17}
\end{align*}
$$

is satisfied for all test functions $v \in H^{1}(\Omega)$. After pulling back the equations to the parameter domain, we look for $u=\hat{u} \circ G$ that satisfy

$$
\begin{align*}
& \int_{[0,1]^{2}} \frac{1}{\epsilon_{c r}} J^{-T} \nabla u \cdot J^{-T} \nabla v \operatorname{det} J \mathrm{~d} x+k_{0}^{2} \int_{[0,1]^{2}} \mu_{r} u v \operatorname{det} J \mathrm{~d} x \\
& \quad+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}} u v\left|\frac{\partial G}{\partial \xi}\right| \mathrm{d} \xi \\
& =\frac{1}{\epsilon_{c r}} \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}}\left(\frac{\partial u^{i}}{\partial n} \circ G+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) u^{i} \circ G\right) v\left|\frac{\partial G}{\partial \xi}\right| \mathrm{d} \xi \tag{3.18}
\end{align*}
$$

| $f$ | $\mu_{r}$ | $\mu_{r}^{s}$ | $\sigma$ | $\epsilon_{0}$ | $\mu_{0}$ | $\epsilon_{r, \text { gold }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $4 \cdot 10^{14}[H z]$ | 1.0 | 1.0 | $10^{6}[S / m]$ | $\left(\mu_{0} c^{2}\right)^{-1}$ | $4 \pi 10^{-7}$ | $-20.199+\mathrm{j} 1.381$ |

Table 3.1: Physical parameters
for all test functions $v=\hat{v} \circ G$. Where $J$ is the Jacobian determinant of the geometry parametrization $G$. As the material parameter $\epsilon_{c r}$ attains a different value in the dielectric (air) and the reflector (gold), we model the physical domain using 5 different patches, one for the reflector and 4 for the surrounding air. The patch layout is sketched in Figure 3.5. We can change the shape opt the reflector by changing the shape of the interfaces between patch 4 (the reflector) and the surrounding patches, we effectively change the shape of the reflector. This means that the control points that defines these interfaces will be used as design variables. Now following the Galerkin method, the weak formulation can be discretized by searching for a multivariate spline $u_{h}=\sum_{i=1}^{n} u_{i} \mathcal{R}_{i}(\boldsymbol{\xi})$ that satisfy the weak form (3.18) for test functions $v=\mathcal{R}_{i}$ for $i=1, \ldots, n$. This can be written as a linear system

$$
A \boldsymbol{u}=(K+M+T) \boldsymbol{u}=\boldsymbol{F}
$$

where $\boldsymbol{u}=\left(u_{1}, \ldots, u_{n}\right)^{T}$ and where the entires in $K, M, T$ and $F$ are given by

$$
\begin{align*}
K_{k l} & =\int_{[0,1]^{2}} \frac{1}{\epsilon_{c r}} J^{-T} \nabla \mathcal{R}_{k} \cdot J^{-T} \nabla \mathcal{R}_{l} \operatorname{det} J \mathrm{~d} \boldsymbol{\xi},  \tag{3.19a}\\
M_{k l} & =-k_{0}^{2} \int_{[0,1]^{2}} \mu_{r} \mathcal{R}_{k} \mathcal{R}_{l} \operatorname{det} J \mathrm{~d} \boldsymbol{\xi},  \tag{3.19b}\\
T_{k l} & =\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}} \mathcal{R}_{k} \mathcal{R}_{l}\left|\frac{\partial G}{\partial \xi}\right| \mathrm{d} \xi,  \tag{3.19c}\\
f_{l} & =\int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}}\left(\frac{\partial u^{i}}{\partial n} \circ G+\left(\mathrm{j} k_{0}+\frac{1}{2 r_{t}}\right) u^{i} \circ G\right) \mathcal{R}_{l}\left|\frac{\partial G}{\partial \xi}\right| \mathrm{d} \xi \tag{3.19d}
\end{align*}
$$

we can write it as a real linear system

$$
\left[\begin{array}{cc}
\Re(A) & -\Im(A) \\
-\Im(A) & -\Re(A)
\end{array}\right]\left[\begin{array}{c}
\Re(\boldsymbol{u}) \\
\Im(\boldsymbol{u})
\end{array}\right]=\left[\begin{array}{c}
\Re(\boldsymbol{f}) \\
-\Im(\boldsymbol{f})
\end{array}\right] .
$$



Figure 3.5: Patch layout. Patch 4 is (the symmetric half of) reflector while the rest is the surrounding air.

As we do not have the exact solution of the PDE, we do not have the exact objective value, but use:

$$
\begin{equation*}
E_{h}\left(\boldsymbol{c}, u_{h}\right)=E\left(G\left([0,1]^{2}\right), u_{h}\right)=\int_{[0,1]^{2}} \delta \circ G\left|\hat{u}_{h}\right|^{2} \operatorname{det} J \mathrm{~d} \boldsymbol{\xi} \tag{3.20}
\end{equation*}
$$

To solve the shape optimization problem we will the two methods described in Chapter 3. The one in section 3.1 is based on solving a sequence of subproblems given as

$$
\begin{array}{ll}
\max _{\boldsymbol{\alpha}} & E_{h}\left(\boldsymbol{c}, u_{h}\right), \\
\text { s.t. } & A \boldsymbol{u}=\boldsymbol{F}, \\
& \boldsymbol{c}=\boldsymbol{c}_{0}+\Delta \boldsymbol{c}, \\
& H_{c^{I}, c^{I}} \Delta \boldsymbol{c}^{\boldsymbol{I}}=-\nabla W_{c^{I}}-H_{c^{I}, c^{B}} \Delta \boldsymbol{c}^{B}, \\
& \boldsymbol{c}^{B}=\boldsymbol{\alpha}, \\
& \boldsymbol{d} \geq \varepsilon, \\
& \boldsymbol{\alpha}_{L} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{U} . \tag{3.21~g}
\end{array}
$$

We will refer to this method as the 'linearization based' method. After solving each subproblem (3.21) we find a new reference parametrization $\boldsymbol{c}_{0}$. We will refer to this as 'reparametrizations', so when we state 'after 5 reparametrizations' it means that we have solved 5 of the subproblems (3.21)The other method we will apply is the one described in section 3.2. With this method we solve a single optimization problem, with the position of all control points entering the formulations as design variables. We include a regularization term to drive the optimization towards high quality parametrizations. The discrete optimization problem is given as

$$
\begin{align*}
\max _{\boldsymbol{c}} & E_{h}\left(\boldsymbol{c}, u_{h}\right)-R_{\tau},  \tag{3.22a}\\
\text { s.t. } & A \boldsymbol{u}=F,  \tag{3.22b}\\
& \boldsymbol{c}_{L} \leq \boldsymbol{c} \leq \boldsymbol{c}_{U} . \tag{3.22c}
\end{align*}
$$

We will refer to this method as the 'regularization based' method. We use design bounds where the design variables, eg. the boundary control points, should be inside a box of width $0.675 r_{t}$ and height $0.475 r_{t}$ centered around $\left(0,0.025 r_{t}\right)$ to allow for a gap between the two reflectors. We will here impose these constraint on the boundary control points, which guarantees that also the boundary curve of the reflector is within these bounds. For the regularization based method we only impose these bounds on the boundary controlpoints, even though the design variables are not limited to the position of the boundary control points. For both methods the position of the control points that control the half circle that defines the truncated domain are fixed. The control points on the boundary at $x=0$, eg. between the two symmetric halves, are
only fixed in the $y$-direction. The fact that we do not fix these in the $x$-direction allows for better parametrizations. ${ }^{2}$

We will consider a few different starting guesses for the shape of the reflector, howeve if not stated otherwise the starting guess will be a circular reflector as sketched in Figure 3.5.

### 3.3.3 Results using Linearizations

In this section we will present the results when using linearization based method described in section 3.1. We will consider two different meshes, a coarse one where the number of degrees of freedom are $N=2548$ and a finer mesh with $N=9300$ degrees of freedom.

As a tolerance for solving the subproblems (3.10) we use tol $=10^{-3}$. We use a fixed number of reparametrizations namely 10 for the coarse mesh and 5 for the fine mesh, as no further progress was seen hereafter.

In Figure 3.6 we show the shape of the reflector at different stages of the optimization process. We see that the designs become more complicated after each reparametrization. This might be since the linearized parametrization methods somewhat limits the shapes that are feasible in each subproblem and therefore you need to reparametrize a few times for the shape to change significantly.

The objective function during the optimization process is shown in Figure 3.7 with a blue line. We see that after each reparametrization, marked with a vertical line, the objective increases until it flattens again. After around 5 or 6 reparametrizations less progress is observed in each subproblem. The fact that the objective flattens during each subproblem suggest that a lower tolerance for the subproblems might reduce the number of iterations needed.

In Figure 3.7 we mark the value of the objective when calculated with a twice uniformly refined mesh. When calculated on this refined mesh the objective is $E_{h / 4}=1.556$ while the objective calculated with the mesh used for the optimization is $E_{h}=1.803$. This is a difference of $16 \%$ which might indicate that the optimization algorithm exploits discretization error to achieve an artificial high objective value.

In Figure 3.8 we plot the designs during the optimization process when using the fine mesh. We see that the final design are qualitatively similar to the one obtained with the coarse grid. However the objective is more accurate as seen from the objective function shown in Figure 3.7. The objective calculated on on a twice uniform refined grid is $E_{h / 4}=1.628$ while the objective calculated with the mesh used for the optimization is $E_{h}=1.638$. This is a $0.6 \%$ difference. The algorithm seem to converge faster when using the fine mesh, as we saw no further progress after 5 reparametrizations.

[^4]

Figure 3.6: The designs at different stages of the optimization process, when using the linearization based method and the coarse mesh. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure 3.7: The objective function during optimization process with the linearization based method for the coarse and fine meshes. The vertical lines indicates where the parametrization is updated. $E_{h / 4}$ is the objective calculated on a refined mesh.

Until now we have only considered one starting guess, namely where the reflector is shaped as a circle. To investigate how the starting guess affects the result, we consider two additional starting guesses. One is where the reflector is shaped as a square while the other one is the final design when using the regularization approach described in section 3.2, with regularization parameter $\tau=0.125$. In Figure 3.9 the starting guesses, the final designs and the objective are shown. We see that we in fact find quite different designs when using the different starting guesses. The objective values of the final designs differs with up to $3 \%$. Shape optimization problems are know to be prone to local optima, so it is not surprising that the final design depends on the starting guess.

Another thing we found while experimenting with the method were that the number of quadrature points used for calculating the Winslow functional can affect the result of the optimization. In the results presented here we use 12 quadrature points for each knot interval in each direction, unless otherwise stated. In Figure 3.10 we show the final designs, after 5 reparametrizations, for a different number of quadrature points. Here the circular reflector from Figure 3.9(a) is used as a starting guess. We see that we end up with different designs in all 4 cases, but with similar objective value.


Figure 3.8: The designs at different stages of the optimization process, when using the linearization based method and the fine mesh.

(a) $E_{h}=0.202$

(d) 5 reparametrizations, $E_{h}=1.638, E_{h / 4}=1.627$, 433 iterations

(g) Objective

(b) $E_{h}=0.114$

(e) 5 reparametrizations, $E_{h}=1.695, E_{h / 4}=1.659$, 328 iterations

(h) Objective

(c) $E_{h}=1.545$

(f) 5 reparametrizations, $E_{h}=1.692, E_{h / 4}=1.698$, 316 iterations

(i) Objective

Figure 3.9: Results for different starting guesses using the fine mesh. In the top row we show the starting guesses, in the middle row the design after 5 reparametrizations and in the bottom row we show the objective value during the optimization process. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure 3.10: The design after 5 reparametrizations, when using a different number of quadrature points pr knot interval in each direction. The fine mesh is used for approximating solutions to the PDE and the circular reflector in Figure 3.9(a) is used as a starting guess.

### 3.3.4 Results using Regularizations

In this section we will present the results for the problem of designing electromagnetic reflectors, when using the regularization based method described in Section 3.2. We will present the final designs and objective for different choices of regularization parameter $\tau$.

In Figure 3.11 the final designs for different values of $\tau$ is presented. From this figure it is clear that the results are sensitive to the value of $\tau$. If $\tau$ is too large then the final design has a low Winslow functional but also a low (poor) objective $E_{h}$, as it is the case in Figure 3.11(a) for $\tau=1 / 4$. On the other hand if $\tau$ is chosen too small the placement of the inner control points will be exploited by the optimization algorithm to find designs with artificially large objective due to discretization error. This is observed in Figure 3.11(b) for $\tau=1 / 32$. Note that in this case the optimization did not even succeed, since the maximum number of iterations were reached. But in between these two extremes there is a sweet spot as seen in Figure 3.11(c) where we have $\tau=1 / 16$. Here the parametrization seems to have a good quality while the objective $E_{h}$ is large. The quality of the parametrization is indicated by the fact that the difference between $E_{h / 4}$ and $E_{h}$ is small, in this case the relative difference is $\left(E_{h}-E_{h / 4}\right) / E_{h / 4}=0.0006$. This indicates that the discretization error is also small. It should be noted that not only the parametrization affects the magnitude of the discretization error, also the field we are approximating affects it. But as a comparison we have $\left(E_{h}-E_{h / 4}\right) / E_{h / 4}=0.53$ for $\tau=1 / 32$, where to parametrization looks of poor quality. In Figure 3.12 we plot the electrical energy $E_{h}$, the regularization term $\tau W$ and the regularized objective $E_{h}-\tau W$.

In Figure 3.13 we show the final designs when using the fine mesh. Importantly


Figure 3.11: The final designs when using the regularization based method and the coarse mesh for different values of $\tau$. ${ }^{1}$ For $\tau=1 / 32$ the optimization algorithm terminated from reaching the maximum number of iterations. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure 3.12: The optimization history when using the regularization based method and the coarse mesh for different values of $\tau$. We plot the electrical energy $E_{h}$, the regularization term and $E_{h}-\tau W$ which is the actual function that is maximized. ${ }^{1}$ For $\tau=1 / 32$ the optimization algorithm terminated from reaching the maximum number of iterations.
we see here that the sweet spot with $\tau=1 / 16$ produce similar results with the fine mesh. But we further observe that $\tau$ can in this case be further reduced without resulting in poor parametrizations, specifically produce good results for $\tau=1 / 64$ as seen in Figure 3.13(b). It can still be too small though, as seen in Figure 3.13(c) for $\tau=1 / 128$.

In Figure 3.14 we show the final designs when using different starting guesses where we use the fine mesh. We have chosen to report these for $\tau=1 / 16$ as it were a sweet spot for both the coarse and fine mesh. Moreover for this value of $\tau$ we actually find a different design when starting from a reflector shaped as a square, as seen in Figure 3.14(e). How ever if $\tau$ is decreased to $\tau=1 / 32$ the same design in was found for all three starting guesses, as seen in Appendix B, Figure B.6.


Figure 3.13: The final designs when using the regularization based method and the fine mesh for different values of $\tau$.


Figure 3.14: Results for different starting guesses when using the regularization based method and the fine mesh. In the top row we show the starting guesses and in the bottom row we show the final designs.

Final designs for more values of $\tau$ can be seen in Appendix B.
In Figure 3.15 we plot the squared magnitude of the field $u$, for 3 different designs; the initial design, the final design when using linearizations and for the design using regularization for $\tau=1 / 32$. The field is calculated on the fine mesh. We see that with both methods the final design of the reflector concentrate energy at the desired point.

### 3.4 Comparison and Discussion

I this section we will compare the two methods in Section 3 based on the results from the model problem of designing electromagnetic reflectors. In Table 3.2 we compare the two methods on the two different meshes, corresponding to two discretization levels.

We see that the objective, when calculated on a refined mesh, $E_{h / 4}$ is quite similar for the two methods on both grids. However when using the linearization based method for the coarse mesh, the objective $E_{h}$ is artificially large, indicating that the optimization likely exploits discretization error. This is seen by the fact the relative difference of $E_{h / 4}$ and $E_{h}$ is $16 \%$. The regularization based approach yields more


Figure 3.15: The field $u$ for three different designs, calculated on the fine mesh.

| Method | Mesh | $\tau$ | $E_{h}$ | $E_{h / 4}$ | $\frac{E_{h}-E_{h / 4}}{E_{h / 4}}$ | Avg time/Iter | Total time |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Lin | coarse | - | 1.803 | 1.556 | $16 \%$ | 9.68 sec | 1.56 h |
| Reg | coarse | $\tau=1 / 16$ | 1.645 | 1.646 | $0.5 \%$ | 11.0 sec | 1.19 h |
| Lin | fine | - | 1.638 | 1.628 | $0.6 \%$ | 53.5 sec | 6.43 h |
| Reg | fine | $\tau=1 / 16$ | 1.646 | 1.646 | $0.006 \%$ | 70.4 sec | 7.20 h |
| Reg | fine | $\tau=1 / 32$ | 1.680 | 1.680 | $0.01 \%$ | 63.8 sec | 6.22 h |

Table 3.2: Comparison of the two methods from Section 3 on two different meshes named 'coarse' and 'fine'. The linearization based method is referred to as 'Lin' while the regularization based method is referred to as 'Reg'. $E_{h}$ is the objective computed on the mesh used in the optimization. $E_{h / 4}$ are the objective computed after refining the mesh uniformly twice. The execution time was measured on a 64 bit HP EliteBook 840 G 4 with and $\operatorname{Intel}(\mathrm{R})$ Core(TM) i7-7500U CPU, with clock rate of 2.70 GHz .
reliable results with respect to this. On the fine mesh the relative differences are smaller, as expected due to the lower discretization error. The regularization based approach seems to produce more reliable results on both refinement levels.

In Table 3.2 we also show the average time per optimization iteration. This time is fairly similar for the two methods. The execution time per iteration is dominated by assembly of the system matrix and the derivatives, which means that, at least for this 2 D problem, the constraints on the determinant in the linearization based method does not increase the execution time significantly. The difference in the average time for the two method is due to a different number of steps taken in the line search part of the optimization algorithm. We experienced that generally more line search steps was necessary for the regularization based method. Also the total running times for the two methods are comparable.

In Section 3.3 .3 we saw that when using the linearization based method different designs were found when using different starting guesses, and also when using different number of quadrature points for calculation the Winslow functional. So it seems that the result is quite sensitive to parameters of the algorithm, which might indicate that there are many local minima. This could be explained by the non linear constraints on the Jacobian determinant, that might introduce additional local optima. For the regularization based approach we generally got more similar results for different number of quadrature points and different starting guesses, which indicates that this method is more robust.

One difference between the two methods that are not apparent from the results, are that the regularization based approach is a lot easier to implement. The key ingredients are the objective function, the Winslow functional and their first order derivatives with respect to controlpoints.

To implement the linearization based method you additionally need the hessian of the Winslow functional, the linearized parametrization method, constraints on the Jacobian determinant, and to setup the outer loop where the subproblems are solved and the design is reparametrized. There are also some more subtle challenges, for example that there is much more bookkeeping as derivatives has to be calculated with respect to inner controlpoints in some cases, and boundary control points in other cases. For this reason the regularization based method could be and easy starting point for investigating a shape optimization problem with IGA.

## Isogeometric

Parameterizations

## using Shape

## Optimization Methods

In section 2 the challenge of finding parametrizations in IGA was presented，and some techniques the finding parametrizations was demonstrated．We saw that the optimization based methods that was able to handle complicated domains，relied on either using constraints on the Jacobian determinant during the optimization or as a mean to generate a valid starting guess．We will introduce the methods for a single patch only，and extend the approach to the multi patch case in Section 4．5．But as covered earlier the number of constraint can become very large especially when going to 3D，making a method like maximizing the smallest spline coefficient of the Jacobian determinant in section 2.2 .5 expensive．In this section we will apply the shape optimization method from section 3.2 to find spline parametrizations for IGA， with the goal of developing a method that do not use explicit constraints and that does not require a valid parametrization as a starting guess．

Let $\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}$ denote the boundary control points of the shape we want to parametrize． Now the main idea is to write the parametrization challenge as a shape optimization problem，where the distance between the boundary control points $\boldsymbol{c}^{\boldsymbol{B}}$ and $\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}$ is minimized：

$$
\begin{align*}
& \min _{\boldsymbol{c}^{B}} \frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{B}\right\|_{2}^{2},  \tag{4.1a}\\
& \text { s.t. } \operatorname{det} J>0, \tag{4.1b}
\end{align*}
$$

which has an optimum at $\boldsymbol{c}^{\boldsymbol{B}}=\boldsymbol{c}_{\text {goal }}^{B}$ with inner control points such that $\operatorname{det} J>0$ ， if there exist a parametrization．We will now use the regularization approach from
section 3.2 on this problem and consider the optimization problem

$$
\begin{equation*}
\min _{\boldsymbol{c}} \frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\mathrm{goal}}^{\boldsymbol{B}}\right\|_{2}^{2}+\tau W(\boldsymbol{c}) \tag{4.2}
\end{equation*}
$$

where we optimize over all control points $\boldsymbol{c}$. Again here we set $W=\infty$ if $\operatorname{det} J \leq 0$ at one of the quadrature points. This optimization problem looks similar to the minimization of the Winslow functional in section 2.3, with the added term $\frac{1}{2} \| \boldsymbol{c}^{\boldsymbol{B}}-$ $c_{\text {goal }}^{B} \|_{2}^{2}$. However the key difference is that here we minimize over all control points, not only the inner control points. That means that while we in section 2.3 needed to start from a valid parametrization with $\boldsymbol{c}^{\boldsymbol{B}}=\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}$, we can here start from a design where $\boldsymbol{c}^{\boldsymbol{B}} \neq \boldsymbol{c}_{\text {goal }}^{B}$. One can therefore avoid the issue of finding a valid initial guess as one can start from a design which is easy to parametrize.

The contribution with the proposed method is twofold. Firstly it serves as a parametrization technique in its own right, and secondly it illustrates the ability of the regularization based shape optimization approach from section 3.2 to find complicated shapes while maintaining a valid parametrization.

### 4.1 Finding an Initial Guess

In this section we will consider different ways to find an initial guess for the proposed method. In principle the initial guess can be chosen arbitrarily, as long as it has a positive Jacobian determinant. For example one could start with the identity map. However since the regularization term require that $\operatorname{det} J>0$ in the quadrature points throughout the optimization process, it can be beneficial to start with an initial guess that are close to the goal, but still being easy to parametrize.

### 4.1.1 Multilinear initial guess

The first method we will consider is to use multilinear interpolation of the corners of the target domain. Examples are shown in Figure 4.1.

However while this method produces a start guess that are close to the final domain, it does not necessarily produce a valid starting guess, i.e., with $\operatorname{det} J>0$. Examples of such cases are shown in Figure 4.1.

### 4.1.2 Affine initial guess

Another method we will consider is to use the best possible affine map as a starting guess. If we let $\boldsymbol{c}^{\text {Id }}$ denote the control points of the identity map when expanded in the spline space used for the geometry map, then we can compute an affine parametrization as

$$
\begin{equation*}
\boldsymbol{c}_{i}=A \boldsymbol{c}_{i}^{\mathrm{Id}}+\boldsymbol{b} \quad \forall i=1, \ldots, n \tag{4.3}
\end{equation*}
$$

where $A \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b} \in \mathbb{R}^{d}$. We can now pose the minimization problem:

$$
\begin{array}{ll}
\min _{A, \boldsymbol{b}} & \frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}, \\
\text { s.t. } \boldsymbol{c}_{i}^{\boldsymbol{B}}=A \boldsymbol{c}_{i}^{\boldsymbol{B}, \mathrm{Id}}+\boldsymbol{b} \quad \text { for } i=1, \ldots n^{\boldsymbol{B}} . \tag{4.4b}
\end{array}
$$

This is a quadratic and convex optimization problem with $d^{2}+d$ unknowns. Examples are shown in Figure 4.1.

### 4.1.3 Outlook

One way to guarantee a valid initial guess is with the following approach. Take a corner of the goal domain where $\operatorname{det} J>0$ and calculate the $d$ directional derivatives of $G$ at this point. Then a parallelepiped can be constructed using these directions. We will not investigate this choice of initial guess further in this work, as the methods mentioned above sufficed in the examples that will be considered here.

### 4.2 Snapping Strategies

After solving the optimization problem (4.2) we end up with a parametrization that which boundary does not match the goal. We let $\boldsymbol{c}^{*}$ denote the minimizer of (4.2). Our general observation is that the smaller the regularization parameter $\tau$ is chosen, the closer the final design will be to the goal. But some action has to be taken to achieve a parametrization with the desired boundary $\boldsymbol{c}_{\text {goal }}^{B}$. We will consider two different approaches in this work.

The first approach we will call "snapping". With this approach you simply take the minimizer $\boldsymbol{c}^{*}$ and set the boundary control points to the goal $\boldsymbol{c}_{\text {goal }}^{B}$ as

$$
\boldsymbol{c}_{i}^{\text {snapped }}= \begin{cases}{\left[\boldsymbol{c}^{*}\right]_{i}} & \text { if } i \text { is an inner CP }  \tag{4.5}\\ {\left[\boldsymbol{c}_{\text {goal }}^{B}\right]_{j}} & \text { if } i \text { is a boundary CP with index } j\end{cases}
$$

where CP is short hand for control point. After the snaping the parametrization need not to be valid, so one has to check whether this is the case. Our general observation is that if $\tau$ is small enough the boundary control points are so close to the goal that the parametrization will remain valid after the snapping.

The second approach we will consider we will call "Winslow snapping". The idea here is that we use the parametrization strategy from section 3.1, where we minimize a linearization to the Winslow functional, to update the inner control points when we do the snapping. Note that this approach is considerably more expensive as we need to compute the Hessian of $W$.

### 4.3 Choice of Regularization Parameter

Again in this application the pressing question now is how to choose the regularization parameter $\tau$. An easy answer is to choose it very small, since we then expect $\| \boldsymbol{c}^{\boldsymbol{B}}-$ $\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}} \|_{2}^{2}$ to be small, and therefore the snapping strategy is more likely to succeed. However as we shall see later we find that often the optimization algorithm needs more iterations to converge when $\tau$ is small. Motivated by this we develop an approach where we start with a large value for $\tau$ and solve a sequence of the problems on the form in (4.2). After each problem we use 'snapping' and check if the resulting parametrization has $\operatorname{det} J>0$, and if not we decrease $\tau$ with the factor $\rho$. We repeat this process until we have found a valid parametrization. The algorithm is illustrated in Figure 4.2. The reason that we include a check of the positivity of $\operatorname{det} J$ in the quadrature points is that this is much cheaper than computing spline coefficients of the Jacobian determinant. As we solve a sequence of optimization problems we use a large tolerance for the optimization too avoid spending too many iterations during one problem.

### 4.4 Scaling

It is sometime beneficial to scale the objective with $1 / \tau$. This is due to the fact that when $\tau$ is small the minimizer will typically also achieve small 2 -norm $\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}$. This means that the objective become very small, and also the gradient becomes small, which can lead to premature termination of the optimization algorithm. So to address this one can consider the problem

$$
\begin{equation*}
\min _{\boldsymbol{c}} \frac{1}{2 \tau}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}+W(\boldsymbol{c}) \tag{4.6}
\end{equation*}
$$

We will only use this approach for the 3D Jigsaw puzzle piece, as it was not necessary for the other examples.

### 4.5 Multipatch Strategies

Until now we have only considered the single patch case, but often a computational domain is split into a collection of patches. In this case one cannot apply this method one patch at a time since we might not know shape of the interfaces between patches.

One way to extend the method to multipatch domains is to treat the control points on the interfaces as inner control points, and treating two control points, on an interface, that should match as one design variable. This means that the interfaces are guaranteed to match after the optimization. However this approach requires that we have a valid initial guess where the interfaces meet. If one for example uses the multilinear initial guess from section 4.1.1, the interfaces are guaranteed to meet.

One can also find the best affine parametrization as in section 4.1.2, where the affine transformation is the same for each patch.

But sometimes one might want to start from an initial guess where the interfaces do not match. To allow this we can modify the method by adding the distance between the interface control points that should match to the objective, such that we minimize

$$
\begin{equation*}
\frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}+\frac{1}{2} \sum_{i, j \text { should match }}\left\|\boldsymbol{c}_{i}-\boldsymbol{c}_{j}\right\|_{2}^{2}+\tau W(\boldsymbol{c}) \tag{4.7}
\end{equation*}
$$

Then when we use snapping we also need to snap the interfaces, for example by setting each set of control points that should match to their mean value.


Figure 4.1: The two approaches for computing initial guesses for some examples. Note that the multilinear interpolation of the corners fail to find a valid initial guess for the second and third example, since the corners lies on a line. Note that for the last example the affine method also fails to find a valid initial guess since the box found has $\operatorname{det} J<0$, so its orientation is wrong.


Figure 4.2: Flowchart of the decreasing $\tau$ approach. In our experiments we use $\tau_{0}=8$ and $\rho=1 / 4$

### 4.6 2D Examples

In this section we will demonstrate the proposed parametrization technique on some examples. We will use a tolerance of $10^{-3}$ for the optimization when using a fixed regularization parameter. When we use the approach from Section 4.3 where a sequence of problems are solved where $\tau$ is decreased between each problem. We will here use a lower tolerance of $10^{-1}$, and the factor of $\rho=1 / 4$ for decreasing $\tau$. The presented are scaled such that the area are equal to 1 . We use 12 quadrature points per knot interval and the Gauss-Legendre quadrature unless otherwise specified.

### 4.6.1 Jigsaw 2

We start off by considering the jigsaw 2 , which was also considered in section 2.4. The initial guess and final design for $\tau=1 / 2$ is shown in Figure 4.3. The results for different values of $\tau$ is reported in Table 4.1. For $\tau \leq 1 / 2$ both our snapping strategies gives a valid parametrization. It is also seen that the smaller $\tau$ is the more number of iterations are needed to reach the tolerance of $10^{-3}$. This is also seen when we plot the objective in Figure 4.4.

In Figure 4.5 we plot the optimization history when using the approach suggested in Section 4.3. Here we solve a sequence of problems in between which we decrease the value of $\tau$. We start with $\tau=8$. We see that with relative few iterations we find a valid parametrization.

The parametrization we found with this method is, not suprisingly, identical to the one found when minimizing the Winslow functional alone in Figure 2.5(e).


Figure 4.3: The initial guess and final design for jigsaw 2 .


Figure 4.4: The objective function during optimization process, for the jigsaw 2 . $E=\frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{B}\right\|_{2}^{2}$.

| $\tau$ | 1 | $1 / 2$ | $1 / 4$ | $1 / 8$ | $1 / 16$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| $\#$ Iters | 78 | 76 | 102 | 115 | 178 |
| $\frac{1}{2}\left\\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}^{\boldsymbol{G}}\right\\|_{2}^{2}$ | 0.757 | 0.0213 | $5.78 \cdot 10^{-3}$ | $1.58 \cdot 10^{-3}$ | $3.84 \cdot 10^{-4}$ |
| $W$ | 3.27 | 3.34 | 3.39 | 3.41 | 3.43 |
| $\operatorname{det} J$ for snapping | $\leq 0$ | $>0$ | $>0$ | $>0$ | $>0$ |
| $\operatorname{det} J$ for Winslow <br> snapping | $\leq 0$ | $>0$ | $>0$ | $>0$ | $>0$ |

Table 4.1: Results from parametrization tests for the jigsaw 2 puzzle piece. The refinement level needed to prove the validity of the parametrization was $k=1$.


Figure 4.5: The objective function during the optimization for jigsaw 2 , when decreasing tau. The dashed lines indicates the points where tau is decreased.

### 4.6.2 Jigsaw 1

We now consider the more difficult jigsaw 1 puzzle piece. Note that in Section 2.4 we were not able to find a valid parametrization without refining the spline space. But with the method proposed in this chapter we are able to find a valid parametrization, as seen from table 4.2. A valid parametrization was found for $\tau \leq 1 / 16$ when using the Winslow snapping and for $\tau=1 / 32$ with regular snapping. This parametrization is shown in Figure 4.6. To prove that the parametrization is valid we need to refine the spline space in which we expand the Jacobian determinant $k=4$ times. So to be fair, the maximization of the smallest spline coefficient in Section 2 might have worked if the we used this refined spline space for expanding the determinant. However the number of constraints would in this case be 21025 .

In Figure 4.8 we use the approach from Section 4.3. After only 160 iterations we find a valid parametrization. For comparison when using the fixed value $\tau=1 / 32$ we needed 898 iterations.


Figure 4.6: The initial guess and final design for jigsaw 1.

| $\tau$ | $1 / 2$ | $1 / 4$ | $1 / 8$ | $1 / 16$ | $1 / 32$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| $\#$ Iters | 359 | 347 | 534 | 541 | 898 |
| $\frac{1}{2}\left\\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}^{\boldsymbol{G}}\right\\|_{2}^{2}$ | 0.213 | 0.0709 | 0.0207 | $6.12 \cdot 10^{-3}$ | $1.59 \cdot 10^{-3}$ |
| $W$ | 6.10 | 6.49 | 6.76 | 6.92 | 7.05 |
| $\operatorname{det} J$ for snapping | $\leq 0$ | $\leq 0$ | $\leq 0$ | $\leq 0$ | $>0$ |
| $\operatorname{det} J$ for Winslow snapping | $\leq 0$ | $\leq 0$ | $\leq 0$ | $>0$ | $>0$ |

Table 4.2: Results from parametrization tests for the jigsaw 1 puzzle piece. The refinement level needed to prove the validity of the parametrization was $k=4$.


Figure 4.7: The objective function during optimization process, for the jigsaw 1. $E=\frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{B}\right\|_{2}^{2}$.


Figure 4.8: The objective function during the optimization for jigsaw 1 , when decreasing $\tau$. The dashed lines indicates the points where $\tau$ is decreased.

### 4.6.3 Jigsaw 1 Uniformly Refined

To be able to compare with the parametrizations found in Chapter 2 we have included the results when considering the jigsaw 1 puzzle piece on a uniformly refined mesh. Here we are able to find valid parametrizations for $\tau \leq 1 / 8$. The number of iterations that are needed becomes quite large when $\tau$ is decreased, with as much as 3150 iterations for $\tau=1 / 32$, as seen in Table 4.3. This behaviour is also seen in Figure 4.10, where we plot the optimization history. For this example the approach where $\tau$ is sequentially decreased is able to find a valid parametrization in 339 iterations.

Again the parametrization we find with this method is similar to the one found when minimizing the Winslow functional alone shown in Figure 2.5(e)


Figure 4.9: The initial guess and final design for jigsaw 1 uniformly refined once.


Figure 4.10: The objective function during optimization process, for the jigsaw 1 uniformly refined once. $E=\frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}$.

| $\tau$ | $1 / 2$ | $1 / 4$ | $1 / 8$ | $1 / 16$ | $1 / 32$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| \#Iters | 710 | 1190 | 1575 | 1936 | 3150 |
| $\frac{1}{2}\left\\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}^{\boldsymbol{G}}\right\\|_{2}^{2}$ | 0.119 | 0.0356 | 0.102 | $3.18 \cdot 10^{-3}$ | $7.42 \cdot 10^{-4}$ |
| $W$ | 5.58 | 5.81 | 5.96 | 6.05 | 6.13 |
| $\operatorname{det} J$ for snapping | $\leq 0$ | $\leq 0$ | $>0$ | $>0$ | $>0$ |
| $\operatorname{det} J$ for Winslow snapping | $\leq 0$ | $\leq 0$ | $>0$ | $>0$ | $>0$ |

Table 4.3: Results from parametrization tests for the jigsaw 1 uniformly refined once. The refinement level needed to prove the validity of the parametrization was $k=1$.


Figure 4.11: The objective function during the optimization for jigsaw 1 uniformly refined once, when decreasing tau. The dashed lines indicates the points where tau is decreased.

### 4.7 3D Examples

### 4.7.1 Water passage

As a 3D example we will consider a water passage from inside a water turbine ${ }^{1}$. The two possible initial guesses and final design from two different viewpoints are shown in Figure 4.12. The result we will present here are with using the multilinear initial guess. However similar results were obtained when using the affine initial guess and the objective function in (4.7). The only difference was that the number of iterations needed to converge for a fixed value of $\tau$ was $50 \%$ larger when using the affine initial guess.

In Figure 4.13 we show the objective during the optimization process for different values of $\tau$ for this example more or less the same number of iterations are used for different values of $\tau$. We also do not save iterations with the approach where we start with a large value of $\tau$ and decrease it, as seen in Figure 4.14, rather the number of iterations increase. The reason for this is that after the first and second subproblem the distance $E=\frac{1}{2}\left\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}_{\text {goal }}^{\boldsymbol{B}}\right\|_{2}^{2}$ have increased compared to the initial guess, due to the large regularization parameter. This behaviour was not seen in the other examples in the chapter, as they all start from a square (or cube) which is a minimizer for the Winslow functional. One way to solve this issue is to use the method presented in Section 3.2.1 however we have not investigated this further.

| $\tau$ | $1 / 2$ | $1 / 4$ | $1 / 8$ | $1 / 16$ | $1 / 32$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| \#Iters | 86 | 106 | 82 | 81 | 81 |
| $\frac{1}{2}\left\\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}^{\boldsymbol{G}}\right\\|_{2}^{2}$ | 0.51 | 0.27 | 0.13 | 0.058 | 0.024 |
| $W$ | 10.36 | 11.03 | 11.82 | 12.64 | 13.41 |
| $\operatorname{det} J$ for snapping | $>0$ | $>0$ | $>0$ | $>0$ | $>0$ |
| $\operatorname{det} J$ for Winslow snapping | $>0$ | $>0$ | $>0$ | $>0$ | $>0$ |

Table 4.4: Results from parametrization tests for the 3D water passage. The refinement level needed to prove the validity of the parametrization was $k=3$.

[^5]

Figure 4.12: The initial guess and final design for the water passage domain.


Figure 4.13: The objective function during optimization process, for the 3D water passage domain


Figure 4.14: The objective function during the optimization for the 3D water passage domain, when decreasing $\tau$. The dashed lines indicates the points where $\tau$ is decreased.

### 4.7.2 3D Jigsaw piece

| $\tau$ | $1 / 2$ | $1 / 4$ | $1 / 8$ | $1 / 16$ | $1 / 32$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| $\#$ Iters | 149 | 103 | 1389 | Failed (1042) | 1153 |
| $\frac{1}{2}\left\\|\boldsymbol{c}^{\boldsymbol{B}}-\boldsymbol{c}^{\boldsymbol{G}}\right\\|_{2}^{2}$ | 0.10 | 0.035 | 0.011 | 0.31 | 0.00078 |
| $W$ | 3.79 | 3.97 | 4.12 | 3.87 | 4.23 |
| $\operatorname{det} J$ for snapping | $\leq 0$ | $\leq 0$ | $\leq 0$ | $\leq 0$ | $\leq 0$ |
| $\operatorname{det} J$ for Winslow snapping | $\leq 0$ | $\leq 0$ | $\leq 0$ | $\leq 0$ | $\leq 0$ |

Table 4.5: Results from parametrization tests for the 3D Jigsaw puzzle piece. For $\tau=1 / 16$ the optimization failed after 1042 iterations with the error message 'Restoration Failed' which means that ipopts restoration phase was unable to find a point acceptable to the line search filter.

As a final example we consider a 3D jigsaw puzzle piece as seen in Figure 4.15(a). For this example we were not able to find a valid parametrization when using a fixed value of $\tau$ as seen in the Table 4.5. Also solving a sequence of subproblems, decreasing $\tau$ did not yield a valid parametrization when using a fixed number of 12 quadrature points per knot interval with the Gauss-Legendre quadrature. As an example the Jacobian determinant is plotted in Figure 4.15(b), after solving 6 subproblems starting with $\tau=1 / 4$ and decreasing it by a factor 4 after each subproblem. Note here that the plotted determinant are without applying any of the snapping strategies, meaning that the parametrization is already invalid before snapping the boundary control points.

One way to get the method to work on this example is to increase the number of quadrature points between each subproblem, when using the approach from Section 4.3. We started with 12 points per knot interval and increased this number by 3 after each sub problem. When starting with the value $\tau=1 / 4$ and decreasing with the factor $\rho=1 / 4$, a valid parametrization was found after the 6th subproblem where $\tau=1 / 4096$. Here the number of quadrature points where 27 per knot interval. The smallest spline coefficient was $2.15 \cdot 10^{-5}$ and we had to refine the spline space where we expand the Jacobian determinant 5 times to find only positive coefficients. At this refinement level the number of coefficients were 24,137,569. The optimization history can be seen in Figure 4.16. For this approach starting with another values of $\tau$, specifically $\tau=8$ did not yield a valid parametrization.

Another way to get the method to work is to use the Gauss-Lobatto quadrature rule discussed in Section 1.3. With this quadrature rule we could keep the number of quadrature points used to evaluate the Winslow functional constant, using 12 points per knot interval. For this quadrature rule starting with $\tau=1 / 4$ yielded a valid parametrization after 7 subproblems. The smallest spline coefficient after refining the spline space where we expand the Jacobian determinant 4 times was $6.6 \cdot 10^{-4}$.

(a) The 3D jigsaw puzzle piece, the domain is mirror symmetric.

(b) Jacobian determinant, after 6 subproblems. The white circle shows the place where $\operatorname{det} J \leq 0$.

Figure 4.15: The 3D jigsaw puzzle piece, and the Jacobian determinant after solving a sequence of 6 subproblems starting with $\tau=1 / 4$ and decreasing it by a factor 4 after each subproblem. We use 12 quadrature points per knot interval to evaluate the Winslow functional.

When using the Gauss-Lobatto quadrature we also found a valid parametrization when starting with $\tau=8$, so the proposed method seems to perform better using the Gauss-Lobatto quadrature.

### 4.8 Conclusion

In this chapter we proposed a parametrization strategy based on the regularization based shape optimization method presented in Section 3.2. This method avoids explicit constraints on the Jacobian determinant.

It performs really well as a 2 D parametrization method, and we were able to find a parametrization of the difficult jigsaw 1 on the coarsest refinement level where the methods in Section 2 failed. Moreover it is very simple to implement as only the Winslow function and its derivatives are needed.

Furthermore the problem (4.2) can be seen as a benchmark shape optimization problem, where the final shape is known. In 2D the results obtained here indicates that this method is able to find very complicated shapes, and a corresponding valid parametrization. So it supports the conclusion from chapter 3 that this is a very promising method for 2D shape optimization with isogeometric analysis.

It was observed that in many cases the number of iterations to find a valid


Figure 4.16: The objective function during the optimization for the 3D jigsaw puzzle piece, when decreasing $\tau$. The dashed lines indicates the points where $\tau$ is decreased. Note that the number of quadrature points is increased by 3 per knot interval after each subproblem.
parametrization could be significantly reduced, when solving a sequence of problems in between which $\tau$ is decreased, compared to using a fixed value of the regularization parameter $\tau$, and at the same time this avoids having to find a suitable value of $\tau$.

As a 3D parametrization method, the results are promising. The method worked well for a 3D water passage domain, where a valid parametrization was found for many different values of $\tau$. We were also able to parametrize a complicated 3D jigsaw puzzle piece. For this domain some tuning of the method was needed, to get the method to work with Gauss-Legendre quadrature. Namely we had to increase the number of quadrature points used to calculate the Winslow functional, and even when decreasing $\tau$ between a sequence of problems, the initial value of $\tau$ affected the outcome. However when using the Gauss-Lobatto quadrature a constant number of quadrature points could be used, and the method performed more robustly. So all in all as a 3D parametrization method the results are promising and support further testing of the method as a shape optimization framework for 3D shape optimization with IGA.

## снатtr 5

## 3D shape optimization with IGA

In this chapter we will consider a 3D shape optimization problem, and use the regularization based approach described in Section 3.2.

The problem we consider is, similar to the 2D problem in Section 3.3, a problem of designing reflectors, but in this case it is in the setting of free surface hydrodynamics. The aim is to find the shape of two opposing reflectors, that maximizes the scattered energy from a plane wave at a point between the two reflectors, as illustrated in Figure 5.1.

### 5.1 Problem Description

### 5.1.1 Governing Equations

We consider an incompressible fluid and assume that it is inviscid. We will consider a domain that extend to infinity in all directions. If we assume the waves to be small we can assume that the free surface located at $z=0$ at all time. A similar setup was considered in [31], with the goal of optimizing an array of cylinders to cloak an object. More details on free surface hydrodynamics can be found in [32]. We assume harmonic time dependency such that the velocity potential $\Phi: \Omega_{\infty} \times \mathbb{R} \rightarrow \mathbb{R}$ is given as

$$
\begin{equation*}
\Phi(\boldsymbol{x}, t)=\Re\left(\phi(\boldsymbol{x}) \mathrm{e}^{\mathrm{j} \omega t}\right), \tag{5.1}
\end{equation*}
$$

where $t$ is the time, $\omega$ is the angular frequency and j is the imaginary unit. The field $\phi$ is complex, and satisfy the Laplace equation

$$
\begin{array}{rlrl}
-\widetilde{\Delta} \phi & =0 & \text { in } \Omega_{\infty}, \\
K \phi-\frac{\partial \phi}{\partial z} & =0 & \text { on } \Gamma_{f}, \\
\frac{\partial \phi}{\partial \boldsymbol{n}} & =0 & \text { on } \Gamma_{\text {Symm }}, \\
\frac{\partial \phi}{\partial \boldsymbol{n}} & =0 & & \text { on } \Gamma_{s}, \tag{5.2d}
\end{array}
$$



Figure 5.1: The setup of the 3D problem. The three squares and the arrow indicates that we consider a plane wave coming from the right of the domain. The blue surface is the surface of the water. Finally the two opposing objects are the reflectors that we want to design.
where $\widetilde{\Delta}$ is the laplace operator and $\Gamma_{f}$ is the free surface at $z=0$. Since the problem is symmetric around $y=0$ we will only optimize over half of the domain. The symmetry boundary at $y=0$ is denoted $\Gamma_{\text {Symm }}$ and here $\phi$ has to satisfy (5.2c). The setup is sketched in Figure 5.2 from the front and from the top.

We will split the field $\phi$ into an incident part $\phi_{I}$ and scattering part $\phi_{S}$ such that

$$
\begin{equation*}
\phi=A\left(\phi_{I}+\phi_{S}\right), \tag{5.3}
\end{equation*}
$$

where $A$ is the amplitude. We will consider the incident field corresponding to a plane wave that decays exponentially in the $z$-direction:

$$
\begin{equation*}
\phi_{I}(x, y, z)=\frac{g}{\omega} \mathrm{e}^{K z-\mathrm{j} K x} . \tag{5.4}
\end{equation*}
$$



Figure 5.2: Infinite domain

The incident wave satisfy

$$
\begin{align*}
\widetilde{\Delta} \phi_{I} & =0 & \text { in } \Omega_{\infty}  \tag{5.5a}\\
K \phi_{I}-\frac{\partial \phi_{I}}{\partial z} & =0 & \text { on } \Gamma_{f}  \tag{5.5b}\\
\frac{\partial \phi_{I}}{\partial \boldsymbol{n}} & =0 & \text { on } \Gamma_{\text {Symm }} . \tag{5.5c}
\end{align*}
$$

Consequently, by inserting $\phi=A\left(\phi_{I}+\phi_{S}\right)$ into (5.2), we see that the scattering field $\phi_{S}$ satisfies the PDE:

$$
\begin{align*}
-\widetilde{\Delta} \phi_{S} & =0 & \text { in } \Omega_{\infty}  \tag{5.6a}\\
K \phi-\frac{\partial \phi_{S}}{\partial d z} & =0 & \text { on } \Gamma_{f}  \tag{5.6b}\\
\frac{\partial \phi_{S}}{\partial \boldsymbol{n}} & =0 & \text { on } \Gamma_{\mathrm{Symm}}  \tag{5.6c}\\
\frac{\partial \phi_{S}}{\partial \boldsymbol{n}} & =-\frac{\partial \phi_{I}}{\partial \boldsymbol{n}} & \text { on } \Gamma_{s} \tag{5.6d}
\end{align*}
$$

We will in this work use the physical parameters $K=3.00 \mathrm{~m}^{-1}, g=9.82 \mathrm{~m} / \mathrm{s}^{2}$ and $\omega=\sqrt{K g}=5.43 \mathrm{~s}^{-1}$

To deal with the infinite domain we will use a technique called Perfectly Matched Layers (PML). Here an absorbing layer $\Omega_{P M L}$ is added around the domain of interest $\Omega_{0}$ as illustrated in Figure 5.3. The details on this formulation can be found in Appendix C.

We need to emphasise that since the goal of this shape optimization problem to increase the wave energy, this model is not the best physical model, as it assumes that the waves are small. So the problem is to be regarded as an academic problem, where the goal is to demonstrate and investigate the approach from Section 3.2 on a PDE-constrained shape optimization problem in 3D.


Figure 5.3: A sketch of the setup for the 3D water reflector. The colors denote different patches, the black lines indicate the interface between the PML layers and the domain of interest. The white lines indicate the part of the boundary that we optimize, namely $\Gamma_{s}$. We let $\Omega=\Omega_{0} \cup \Omega_{P M L}$ denote the full domain.

### 5.1.2 Convergence test

To test the PML formulation we will consider a problem where the solution is known. Specifically we consider the PDE (5.2) but replacing the Neumann boundary condition at $\Gamma_{s}$ with a Dirichlet boundary condition

$$
\phi=\phi_{\text {exact }} \quad \text { on } \Gamma_{s} .
$$

We will also change the domain of interest such that $\Gamma_{s}$ has center at $(0,0,0)$. A solution to this problem is given by

$$
\begin{equation*}
\phi_{\text {exact }}=e^{K z}\left(\mathcal{J}_{0}\left(K \sqrt{x^{2}+y^{2}}\right)+\mathrm{j} \mathcal{Y}_{0}\left(K \sqrt{x^{2}+y^{2}}\right)\right) \tag{5.7}
\end{equation*}
$$

where $\mathcal{J}_{0}$ is the 0th Bessel function of first kind and $\mathcal{Y}_{0}$ is the 0th Bessel function of the second kind. To implement the Bessel functions $\mathcal{J}_{0}$ and $\mathcal{Y}_{0}$ we use the power
series

$$
\begin{align*}
& \mathcal{J}_{0}(z)=\sum_{i=0}^{\infty}(-1)^{k} \frac{\left(1 / 4 z^{2}\right)^{k}}{(k!)^{2}}  \tag{5.8}\\
& \left.\mathcal{Y}_{0}(z)=\frac{2}{\pi}(\ln (1 / 2 z)+\gamma) \mathcal{J}_{0}(z)+\frac{2}{\pi} \sum_{i=1}^{\infty}(-1)^{n+1}\left(1+\frac{1}{2}+\cdots+\frac{1}{n}\right) \frac{1}{(n!)^{2}}\left(1 / 4 z^{2}\right)^{n}\right) \tag{5.9}
\end{align*}
$$

where the Euler constant is given by $\gamma=0.57721566490153286060$ [35]. We truncate the series at $N=50$. Note that numerical evaluation of the Bessel functions by this power series is not the most effective method, nor is it numerical stable for large values of $r=\sqrt{x^{2}+y^{2}}$. We here use these power series as they are easy to implement and since they are only used to test the implementation. For a more efficient method, see for example [21].

We will use splines of degree $p=2$ and inner knots $0,0.5,1$ to parametrize each patch of the domain $\Omega$. We will approximate the solution to the PDE on three different refinement levels $r=0, \ldots, 2$. At refinement level $r$ use the spline space used for representing the geometry refined $r$ times uniformly, to approximate the solution $\phi_{h}$. The largest knot interval is of size $h=0.5, h=0.25$ and $h=0.125$ respectively.

The results at refinement level 2 is shown in Figure 5.4. We see that the waves decay in the PML regions, and the difference between $\phi_{h}$ and $\phi_{\text {exact }}$ is small, except where there is a singularity for $\mathcal{Y}_{0}$ at $x=y=0$.

In Figure 5.5 we plot the error of the real and imaginary part at different levels of refinement. We use splines of degree $p=2$. Since we ran the code on a laptop we were not able to decrease $h$ further that $h=0.125$, which is necessary to verify the convergence rate. That said, it seems that the real and imaginary part converge. We see that the imaginary part converge slower that the real part, however since there is a singularity at $x=y=0$ for the imaginary part, we do not expect the same rate of convergence.

We will use the refinement level 1 and 2 when performing shape optimization.


Figure 5.4: The solution to the test equation compared with the exact solution, at refinement level 2 . There is a singularity for $\Im\left(u_{\text {exact }}\right)$ at $x=y=0$


Figure 5.5: The $L_{2}$-error $\sqrt{\int_{\Omega_{0}} \Re\left(\phi_{h}-\phi_{\text {exact }}\right)^{2} \mathrm{~d} V}$ and $\sqrt{\int_{\Omega_{0}} \Im\left(\phi_{h}-\phi_{\text {exact }}\right)^{2} \mathrm{~d} V}$ against the mesh size, for the three refinement levels: 0,1 and 2 . The expected rate of convergence $O\left(h^{3}\right)$ is plotted as a reference.

### 5.2 Shape Optimization Framework

We will consider the shape optimization problem

$$
\begin{align*}
& \max _{\boldsymbol{c}} E_{h}  \tag{5.10a}\\
& \text { s.t. } \boldsymbol{c}_{L} \leq \boldsymbol{c} \leq \boldsymbol{c}_{U} \tag{5.10b}
\end{align*}
$$

with objective

$$
\begin{equation*}
E_{h}=\frac{1}{2} \int_{\Omega} \delta\left|\phi_{h}\right|^{2} \mathrm{~d} x \tag{5.11}
\end{equation*}
$$

where $\phi_{h}$ is the spline approximation of the solution to the PDE. The weight function $\delta$ is given by the Gaussian function

$$
\begin{equation*}
\delta=\frac{1}{\sigma^{3}(2 \pi)^{(3 / 2)}} \mathrm{e}^{\frac{x^{2}+y^{2}+z^{2}}{2 \sigma^{2}}} \tag{5.12}
\end{equation*}
$$

where the constant $a$ is chosen such that $\int_{\mathbb{R}^{3}} \delta \mathrm{~d} V=1$. We will use $\sigma=0.1$. The weight function $\delta$ is plotted in Figure 5.6. In this Figure we also show the initial design and the bounding box of $\Gamma_{s}$. We use bounds so that all the control points of $\Gamma_{s}$ has to be inside this bounding box. In the z-direction the controlpoints can be between 0 and -0.5 , for reference the PML region starts a $z=-1$ and ends at $z=-2$. We also bound each side such that its x and y coordinate does not pass the x and y coordinate of the center of the initial square. This was in an attempt to ensure that opposing sides would not intersect. We will fix the parametrization of the


Figure 5.6: The weight function $\delta$. The initial design is marked with white. The bounding box of the control points of $\Gamma_{s}$ is marked with red.

PML patches, mainly to avoid to differentiate the PML terms with respect to control points, and to ensure that the optimization cannot exploit a poor parametrization in the PML region such that energy is reflected back into the domain. This also reduces the number of optimization variables.

We will apply the method for shape optimization based on regularization described in 3.2. While experimenting with this optimization problem we found that the boundary $\Gamma_{s}$ would self intersect after the optimization. This can happen without any of the parametrizations becoming invalid and is a well known challenge in shape optimization with IGA [33]. One possibility is to derive self intersection constraints as in [33]. However in this work we will instead parametrize the interior of $\Gamma_{s}$ and maintain a parametrization $G_{s}$ via the same regularization approach. This is done by adding an extra regularization term

$$
\begin{align*}
& \min _{\boldsymbol{c}, \boldsymbol{\alpha}}-E_{h}+\tau\left(W(G)+W\left(G_{s}\right)\right)  \tag{5.13a}\\
& \text { s.t. } \boldsymbol{c}_{L} \leq \boldsymbol{c} \leq \boldsymbol{c}_{U} \tag{5.13b}
\end{align*}
$$

where $\boldsymbol{\alpha}$ are coordinates for the control points of $G_{s}$.
To avoid confusion we shall avoid using the term objective function, but rather refer to $E_{h}$ as the energy, and to $\tau\left(W(G)+W\left(G_{s}\right)\right)$ as the regularization term.

Another thing we experienced, as will be demonstrated in the results section 5.3, was that the parametrization became invalid on the boundary or interfaces. To combat this we will try to use Gauss-Lobatto quadrature to compute the Winslow functional. This quadrature rule is a closed rule, meaning that it includes values at the end points of the integration interval.

### 5.3 Results

In this section we will go through the results when applying the regularization based approach to this problem. We will show results for two different quadrature rules, the Gauss-Legendre quadrature rule, and the Gauss-Lobatto quadrature that includes function evaluations at the endpoints of the interval.

We will use degree $p=2$ and inner knots $0, \frac{1}{2}, 1$ in all directions for representing the geometry. We will consider two different refinement levels for solving the PDE. At refinement level $r$ we use the spline space used for representing the geometry refined uniformly $r$ times. We will consider the cases $r=1$ and $r=2$, i.e., the inner knots are $0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$ and $0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1$, respectively.

### 5.3.1 Gauss-Legendre Quadrature

### 5.3.1.1 Minimizing Winslow

In our first experiment we disregard the energy $E_{h}$ and minimize the Winslow function alone. This corresponds to letting $\tau \rightarrow \infty$. The design that minimizes the Winslow functional, when calculated with the Gauss-Legendre quadrature using $n_{Q}=12$ quadrature points per knot interval, is shown in Figure 5.7(a). The geometry map found here is not a valid parametrization. This is since the Jacobian determinant becomes negative in two corners of patch 4, as seen in Figure 5.7(b). Since it is only
negative very close to the corner, and since the Gauss-Legendre quadrature is open, it is not negative in a quadrature point. Even though we are interested in minimizing a combination of the energy and Winslow, this result is concerning, since we cannot expect to get a valid parametrization even if $\tau$ goes to $\infty$. It also indicates that using a closed quadrature rule might be better, since it includes values at the corners.

(a) The final design. Patch 4 is yellow.

(b) The Jacobian determinant on patch 4. The places with negative determinant are marked with white circles.

Figure 5.7: The design that minimizes Winslow. For this design we have $W=86.2145$. It has negative determinant in two of the corners of patch 4 . Note that the color map only covers the values around 0 .

### 5.3.1.2 Water Reflector

When we use our method for solving the problem (5.13) we found that the Jacobian determinant would be negative for all the values of $\tau$ we tested ( $\tau=0.0625,0.25,1,4,16,64$ ). In Figure 5.8(a) we show the design found after 50 optimization iterations, when using refinement level 1 and regularization parameter $\tau=1$, as an example. Here the Jacobian determinant is negative in some places, as seen in Figure 5.8(b) where we plot the Jacobian determinant for patch 0 and patch 4.


Figure 5.8: The design after 50 iterations for refinement level 1 and $\tau=1$. It has negative determinant for example at patch 0 and 4 . Note that the color map only covers only values around 0 .

### 5.3.2 Gauss-Lobatto Quadrature

We will here show the results obtained when using the Gauss-Lobatto quadrature rule to calculate $W$. Again using $n_{Q}=12$ quadrature points per knot interval.

### 5.3.2.1 Minimizing Winslow

When minimizing the Winslow functional alone we, as expected, get a design that is almost identical to the one in Figure 5.7(a), as seen in Figure 5.9(a). The value of $W$ is only $0.00081 \%$ larger, however this time the Jacobian determinant is positive. Its smallest spline coefficient is $\min \boldsymbol{d}=0.0013$.

### 5.3.2.2 Water Reflector

When using the Gauss-Lobatto quadrature to calculate the Winslow functional, we find a design with a valid parametrization after 50 iterations of the optimization algorithm. The squared magnitude of the velocity potential for the initial design and the design after 50 optimization iterations is shown in Figure 5.10. To prove that the parametrization is valid, the spline space in which the determinant is expanded was refined 4 times to obtain positive coefficients. Its smallest spline coefficient is $5.91 \cdot 10^{-6}$. The energy after 50 iterations is around 50 times larger than for the initial design. As mentioned before we maintain a parametrization of the interior of the reflector, to avoid that the boundary $\Gamma_{s}$ fold over it self. This parametrization is shown in Figure 5.11, the parametrization of the center is also valid, with smallest

(a) The final design. Patch 4 is yellow.

(b) The Jacobian determinant on patch 4

Figure 5.9: The design that minimizes Winslow. For this design we have $W=86.2152$. It has positive determinant. Note that the color map only covers the values around 0 .
spline coefficient of 0.0735 . Also we see that on the bottom, the reflector is shaped as a square. For this problem both the velocity potential and the weight $\delta$ decays exponentially in the z-direction. It seem that the bottom of the reflector is affected more by the Winslow functional, while the shape at the free surface $z=0$ affects the velocity field more.

However later in the optimization the parametrization becomes invalid. The design and a spot with negative determinant is shown in Figure 5.12.

In Figure 5.13 we plot the optimization history. The optimization algorithm did not converge, rather it stopped due to a time limitation of 72 hours, on the server on which the code was run. We see that the energy still increases after 500 optimization iterations. The energy has increased with more than a factor 300 , which means that it might be hard to reach a balance between the Winslow functional and the objective. That said, the regularization term is in this case still several times larger than the energy, so it is unclear why the parametrizations become invalid, and it could very well be that the optimization exploits the places where the determinant is very small, to create an artificially large energy.


Figure 5.10: The energy $\left|\phi_{S}\right|^{2}$ for the initial design and at iteration 50 for refinement level 1 and $\tau=1$.


Figure 5.11: The shape and parametrization of the reflector, from 3 different view points. The red lines are the image of the knot lines. Note that we do not force the controlpoints on the upper boundary to have $z=0$ as we only parametrize the reflector to keep the boundary $\Gamma_{s}$ from self intersecting.

(a) The design after 100 iterations. Patch 0 is white.

(c) The design after 500 iterations. Patch 4 is yellow.

(b) The Jacobian determinant on patch 0 .

(d) The Jacobian determinant on patch 4.

Figure 5.12: The design after 100 and 500 iterations for refinement level 1 and $\tau=1$, using Gauss-Lobatto quadrature. It has negative determinant on patch 0 after 100 iterations, and on patch 4 after 500 iterations. The locations where $\operatorname{det} J<0$ are indicated with a white circle. Note that the color map only covers the values around 0.


Figure 5.13: The optimization history for refinement level 1 and $\tau=1$.

### 5.3.3 Refinement level 2

We will now consider some of the results on a finer discretization, the refinement level 2. Here we were only able to perform 57 iterations in the, 72 hours that the code were running. We will report the results after 50 iterations, to be able to compare with the results on refinement level 1 in Figure 5.10. We consider two different values of $\tau$ namely $\tau=1$ and $\tau=0.25$. In Figure 5.14 we plot the squared magnitude of the velocity potential after 50 iterations. For both values of $\tau$ the parametrization obtained is valid. For $\tau=0.25$ we had to refine the spline space in which we expand the Jacobian determinant 4 times to prove that the parametrization was valid. The smallest spline coefficient was $1.65 \cdot 10^{-4}$. The number of spline coefficients at this refinement level is 96,916 for all patches and 82,275 for the domain of interest. For $\tau=1$ we only had to refine the spline space in which we expand the Jacobian determinant 2 times to prove that the parametrization was valid. The smallest spline coefficient was $1.67 \cdot 10^{-3}$. The number of spline coefficients at this refinement level is 24,878 for all patches and 10,237 for the domain of interest $\Omega_{0}$.

In Figure 5.15 we show the shape of the reflector for the two values of $\tau$. The shapes are similar but not identical, but bear in mind that the optimization did not converge. However we still found an increase in the objective from 0.067 to 4.75 and 5.77 for $\tau=0.25$ and $\tau=1$ respectively. It might be surprising that a higher energy is found for higher value of $\tau$. However note that the optimization did not converge, so a balance between the energy and the regularization term are not achieved. One explanation could be that in Chapter 4 we saw that sometimes convergence is faster for larger $\tau$ which might explain the difference.


Figure 5.14: The energy $\left|\phi_{S}\right|^{2}$ for the initial design and at iteration 50 for refinement level 2.

(a) $\tau=0.25$, seen from above.

(d) $\tau=1$, seen from above.

(e) $\tau=1$, seen from the side.

(f) $\tau=1$, seen from below.

Figure 5.15: The shape and parametrization of the reflector at iteration 50 for $\tau=0.25$ and $\tau=1$, from 3 different view points. The red lines are the image of the knot lines. Note that we do not force the controlpoints on the upper boundary to have $z=0$ as we only parametrize the reflector to keep the boundaries from folding over.


Figure 5.16: The optimization history for refinement level 2 and $\tau=0.25$.

### 5.4 Discussion

We have in this section applied the regularization based shape optimization approach to the 3D shape optimization problem of designing reflectors for free surface flow. We were with the method able to improve the energy with up to a factor 86, while generating a valid parametrization at the same time. However to get this result we had to stop the optimization prematurely, after 50 optimization iterations. After more iterations, the Jacobian determinant becomes negative between the quadrature points used to evaluate the Winslow functional. The negative determinant were mainly seen on the boundary $\Gamma_{s}$ or on the patch interfaces, however, although not shown here, we did find cases where the Jacobian determinant were negative in the interior of the domain.

With the regularization based approach one looks for a shape and parametrization where a balanced between the energy and the Winslow functional is attained, i.e., a shape with high energy and a good quality parametrization. However for this problem after 500 iterations the calculated energy were up to a factor 300 greater than for the initial guess. This observation indicates that it might be hard to find a balance between the energy and the regularization term. That said it was already after between 50 and 100 optimization iterations that the parametrization became invalid and here the regularization term were still several times larger that the energy for some of the values of $\tau$ that we tested.

For the designs with valid parametrizations that we found, we had to refine the space where we expand the Jacobian determinant several times to prove that these parametrizations were valid. In the worst case we ended up with 82,275 spline coefficients. This can be due to different scenarios. Maybe the shape found by the algorithm is indeed very difficult to parametrize. In that case the proposed method performs well, in that it is able to find shapes that would require many constraints on the Jacobian determinant to be found with a constraint based method. However it could also be that we find a shape where minimization of the Winslow functional is a poor parametrization strategy, in which case we might be able to find the same shape by using another parametrization strategy coupled with constraints on the Jacobian determinant. Finally it can also be that the optimization exploits the small Jacobian determinant to introduce numerical errors that artificially increase the energy.

To avoid having the boundary $\Gamma_{s}$ self intersect, we had to maintain a parametrization of the interior of the reflector, even though the PDE was not defined on this domain. This was not a major issue with our setup, since it did not increase the number of optimization variables significantly. This illustrates that for more complex topology additional regularization has to be considered to avoid self intersections. For example self-intersection constraints similar to the ones used in [33].

If one wants to apply this method for a shape optimization problem, then it might be important first to investigate what happens when the Winslow functional is minimized alone, since this can indicate whether additional regularity has to be enforced.

We observed that the while the shapes we found had a complex shape near the free
surface they all seemed to be close to a square lower in the domain. This might be due to the fact that both the weight function $\delta$ and the waves decay exponentially in the z-direction and therefore the Winslow regularization term will dominate away from the free surface. One idea would be to let the regularization parameter vary spatially. For example have a large value at the surface and lower value when $z$ decrease.

We also saw that the results of the approach depended a lot on the quadrature rule used to calculate the Winslow functional. When using the Gauss-Legendre quadrature the Jacobian determinant would become negative on patch boundaries. It seems that the Gauss-Lobatto quadrature is a better choice for this application. However it is concerning that different quadrature rules give completely different results, as it indicates that the Winslow functional is not approximated well. One can increase the number of quadrature points, but this can decrease the efficiency of the method. It would be interesting to try to use an adaptive quadrature rule, such that we can approximate the integrals better without introducing two many quadrature points. One way to do this is to use different number of quadrature points on different knot intervals. There are other alternatives in the literature, see for example [18, 16].

### 5.5 Conclusion and outlook

The proposed method for 3D shape optimization using regularization shows potential, as we were able to increase the energy with a factor 86 and obtain a valid parametrization by solving one optimization problem, without additional constraints. However this was at at intermediate state of the optimization, and after more than 100 iterations it seems to produce designs with negative Jacobian determinant. It seems that the method is sensitive the specific choice of quadrature rule use to compute the Winslow functional.

These results motivate several further investigations. Firstly it would be interesting to compare this framework to a method based on constraints on the Jacobian determinant. This could for example be using the spring method as the parametrization strategy with constraints on the spline coefficients of the determinant if the large number of constraints on the Jacobian determinant are manageable. We unfortunately did not have time to investigate this in this work.

Secondly it could be interesting to investigate this method using an adaptive quadrature rule to compute the Winslow functional, such that the integral in the areas where the determinant is small is still well approximated.

Thirdly an important question is whether it is possible to develop a better regularization term than the Winslow functional. It would for example be interesting to use an error estimate as an additional regularization term as this might guard against exploitation of the numerical errors.

Finally it would be interesting to take the results found in this section, and try to find a parametrization using another 3D parametrization strategy for example the method proposed in [39] and compare the outcome with the one found here. Perhaps
some of the designs with an invalid parametrization we found, can be parametrized using another method.

## chapter 6

## Conclusion and outlook

### 6.1 Conclusion

In this work we have investigated methods of maintaining valid parametrizations for IGA during a shape optimization process, with the aim of developing methods applicable for efficient 3D shape optimization with IGA.

We have considered two different approaches, namely a boundary driven approach based on using constraints on the positivity of the Jacobian determinant and a regularization based approach where the inner control points enter the formulation as additional design variables. We found that for a 2 D model problem of designing electromagnetic reflectors, the regularization based approach performed comparably to the state of the art constraint based method. Furthermore it seemed to produce more reliable results on coarse meshes, while being significantly easier to implement.

Formulating the challenge of finding parametrizations as a shape optimization problem, we further demonstrated that the regularization based approach can be used to find parametrizations of complicated domains. This method worked really well in 2D as it was able to parametrize a difficult jigsaw puzzle piece on a coarse refinement level where the other parametrization techniques we have investigated failed. Additionally it avoids the explicit constraint on the Jacobian determinant.

We also demonstrated that the method could produce valid parametrizations for complicated 3D domains, as we were able to parametrize a 3D jigsaw puzzle piece. There are not many 3D parametrization techniques in the literature, and the proposed method is promising and simple to implement.

Finally we investigated the regularization based approach on a 3D shape optimization problem of designing reflectors for free surface waves. We were able to improve the objective by a factor 86 compared to the initial guess, while still obtaining a valid parametrization of the domain. However this was at an intermediate stage of the optimization, and in several cases the parametrization became invalid if more than 50 iterations of the optimization process was taken.

### 6.2 Outlook

One of the issues with the regularization based approach is how to choose the regularization parameter. When using the method for finding parametrizations in chapter 4 we investigated an approach where a sequence of problems with decreasing regularization parameter are solved. This method did not only provide a way to choose the regularization parameter, but we also found that it could reduce the number of optimization iterations needed to find a valid parametrization. This motivates the investigation of a similar approach for PDE-constrained shape optimization problems. When finding parametrizations the procedure can be stopped when a valid parametrization is found, however for PDE-constrained optimization problems it is not clear what an appropriate stopping criterion should be. One idea would be to estimate the discretization error between each subproblem and stop if it increases above some threshold, indicating that the optimization might exploit the position of inner control points.

For the 3D shape optimization problem considered here, the proposed method would produced parametrizations with small Jacobian determinant, or even invalid parametrizations. It would be interesting to try to reparametrize these shapes with other methods, as this would indicate whether the Winslow functional is a good quality measure in 3D. It would also be interesting to investigate a hybrid approach, where the inner control points are determined by a parametrization technique, but where one uses the regularization term instead of the explicit constraints on the Jacobian determinant.

We found that the results from the shape optimization were sensitive to the quadrature rule used for evaluating the Winslow functional. The constraint based method in section 3.1 were sensitive to the number of quadrature points used to evaluate the Winslow functional for our 2D model problem. For the 3D model problem we saw that the Gauss-Legendre quadrature produced invalid parametrizations, while the GaussLobatto quadrature could produce valid parametrizations. These results encourage the investigation of other, perhaps adaptive, quadrature rules for the evaluation of the Winslow functional. This could for instance be accomplished by identifying the knot intervals where the Jacobian determinant is small and increasing the number of quadrature points used here.

Another question is whether there are better regularization terms. In 2D the Winslow functional performs very well, but it is not clear whether the 3D generalization considered here is the best choice. One idea is to use an error estimate as an additional regularization term, to drive the optimization towards a parametrization that allow for an accurate analysis.

## appenox A

## Paper: "Practical isogeometric shape optimization: <br> Parameterization by <br> means of <br> regularization"

This chapter contains the manuscript: Limkilde, A., Evgrafov, A., and Gravesen, J., Mantzaflaris A. "Practical isogeometric shape optimization: Parameterization by means of regularization", submitted to the Journal of Computational Design and Engineering, 2020.

# Practical isogeometric shape optimization: Parameterization by means of regularization 

A. Limkilde ${ }^{1, *}$, A. Evgrafov ${ }^{2}$, J. Gravesen ${ }^{1}$, and A. Mantzaflaris ${ }^{4}$<br>${ }^{1}$ Department of Applied Mathematics and Computer Science, Technical University of Denmark<br>${ }^{2}$ Department of Mechanical Engineering, Technical University of Denmark<br>${ }^{4}$ Inria Sophia Antipolis, Université Côte d'Azur<br>* Corresponding author, asgl@dtu.dk


#### Abstract

Shape optimization based on Isogeometric Analysis (IGA) has gained popularity in recent years. Performing shape optimization directly over parameters defining the CAD geometry, such as for example the control points of a spline parametrization, opens up the prospect of seamless integration of a shape optimization step into the CAD workflow.

One of the challenges when using IGA for shape optimization is that of maintaining a valid geometry parametrization of the interior of the domain during an optimization process, as the shape of the boundary is altered by an optimization algorithm. Existing methods impose constraints on the Jacobian of the parametrization, to guarantee that the parametrization remains valid. The number of such validity constraints quickly becomes untractably large, especially when 3D shape optimization problems are considered.

An alternative, and arguably simpler approach is to formulate the isogeometric shape optimization problem in terms of both the boundary and the interior control points. In order to ensure a geometric parametrization of sufficient quality a regularization term, such as the Winslow functional, is added to the objective function of the shape optimization problem.

We illustrate the performance of these methods on the optimal design problem of electromagnetic reflectors and compare their performance. Both methods are implemented for multipatch geometries, using the IGA library G+Smo and the optimization library Ipopt. We find that the second approach performs comparably to a state of the art method with respect to both the quality of the found solutions and computational time, while its performance in our experience is more robust for coarse discretizations.


## 1 Introduction

Isogeometric analysis (IGA) introduced in [13] is a Galerkin method that uses splines to approximate both the geometric domain and solutions to partial differential equations (PDEs). Splines are commonly used in computer aided design (CAD) and IGA is an attempt to bridge the gap between simulation and design [3]. This makes it beneficial for shape optimization as the optimization can be performed directly over parameters defining the CAD geometry, for example the control points of a spline parametrization, and it opens up the prospect of seamless integration of a shape optimization step into the CAD workflow.

One of the key challenges when using IGA in general, is that one needs a parametrization of the interior of the physical domain, on which the PDE is posed [21, 12, 9, 16]. This parametrization
is used to pull back the weak form of the PDE to the parameter domain where the basis splines (B-splines) are defined. The choice of parametrization can affect the accuracy of the resulting IGA discretization [9, 25] and at the very least the parametrization should be valid (a bijective map), that is its Jacobian determinant should be non-zero. One approach to constructing a valid parametrization in 2D is to search for the one whose inverse is harmonic. In [12] this property is reformulated as a nonlinear PDE and the parametrization is found by solving this PDE. In [11] this PDE based parametrization technique is used for a gradient based shape optimization algorithm with IGA. In $[9,17]$ the same property is attained by minimizing the Winslow functional [24]. The method can be made more flexible by the use of adaptive splines [4], that allow to enrich the feasible region near to complex boundaries. In the recent works [21, 19] the approach of parametrizing a complex domain by deforming a given template is explored. In [18] the focus is on producing parametrizations with low-rank with respect to the coefficient tensor.

When using IGA for shape optimization the challenge of finding a valid parametrization is even more important, since the shape of the physical domain changes during an optimization process. This means that a valid parametrization needs to be maintained during this process.

To guarantee that the parametrization remains valid during the optimization process, shape optimizations methods based on IGA often rely on constraints on the Jacobian of the parametrization $[7,17]$. These can be enforced either by using injectivity cones or by using the spline coefficients of the Jacobian determinant [26]. However the number of constraints needed quickly becomes very large, especially in 3D. Furthermore when using the coefficients of the Jacobian determinant for the constraints, as we will do in this work, it may be necessary to expand the Jacobian determinant on a finer spline space, which increases the number of constraints even further.

In this work we will compare the existing approach to IGA shape optimization, relying upon reparametrizing the domain, with a simple approach to maintaining a valid parametrization without the use of explicit validity constraints. Namely, one lets the positions of all the control points that defines the parametrization and the shape of the domain enter the formulation as independent optimization variables and adds a regularization term to drive the optimization towards designs with a valid parametrization. Such an approach has been considered in the context of shape optimization in mechanics in [20], and has to the best of our knowlegde, only been considered very briefly in the context of shape optimization with IGA in [6]. It remains a question whether this approach performs comparably to state of the art methods and the aim of the work is to investigate exactly this question.

In this work we illustrate that this simple approach is able to handle complicated geometries, by comparing its performance to a state of the art shape optimization approach based on using a linearization of the Winslow minimization problem as a parametrization strategy, and employing locally refined splines to represent the Jacobian determinant. The method closely resembles the one in [17]. The main difference is that we use Truncated Hierarchical Basis splines (THB-splines) [8], which possess the partition of unity property, to refine the spline space in which the determinant is expanded locally. This reduces the number of constraints needed compared with tensor product global refinement.

We will apply the two methods and compare their performance on the shape optimization problem of designing electromagnetic reflectors. In this problem we have two metallic reflectors in a dielectric medium and search for a shape that maximizes the electrical energy close to a chosen point. The same problem has been studied with topology optimization in [1, 23, 2], and with IGA in [17].

The methods are implemented for multipatch geometries, using the IGA library G+Smo ${ }^{1}$ and the optimization library Ipopt ${ }^{2}$.

The paper is organized as follows. In section 2 we outline the relevant notation, and in sections 3 and 4 we describe the two methods we are going to compare. In section 5 we apply these methods

[^6]to the aforementioned shape optimization problem, and discuss and compare the performance of the two approaches. We end the paper with some discussion and conclusions. Some of the more technical details are presented in the appendices.

## 2 Preliminaries and notation

Let us consider the following PDE-constrained shape optimization problem:

$$
\begin{align*}
& \max _{\Omega \in \mathscr{O}_{a d}} E(\Omega, u),  \tag{1a}\\
& \text { s.t. } a_{\Omega}(u, v)=\ell_{\Omega}(v) \quad \text { for all } v \in V, \tag{1b}
\end{align*}
$$

where $\mathcal{O}_{a d}$ is a set of admissible shapes, $E$ is the objective and (1b) is the governing PDE in the weak form.

Within the IGA framework both $\Omega$ and $u$ will be approximated numerically using splines. Namely, we have $\Omega=G(] 0,1\left[{ }^{d}\right)$, with the parametrization

$$
\begin{equation*}
G(\boldsymbol{\xi})=\sum_{i=1}^{N^{g}} \boldsymbol{c}_{i} R_{i}^{g}(\boldsymbol{\xi}) \tag{2}
\end{equation*}
$$

where $\boldsymbol{c}_{i} \in \mathbb{R}^{d}$ are the control points, $N^{g}$ is the number of control points, and $R_{i}^{g}$ are the basis (B) splines. In this work, unless specifically stated otherwise, we will utilize tensor product B-splines. The superscript $g$ indicates that the B-splines $R_{i}^{g}$ (that is, their degrees and knotvectors) are specific to the geometry representation. Within the shape optimization framework it will sometimes be necessary to distinguish between boundary and inner control points. We will therefore introduce the notation

$$
\boldsymbol{c}=\left[\begin{array}{l}
\boldsymbol{c}^{\notin}  \tag{3}\\
\boldsymbol{c}^{i}
\end{array}\right]
$$

where $\boldsymbol{c}^{t}$ are the boundary control points and $\boldsymbol{c}^{i}$ are the inner control points. The Jacobian $J=\frac{\partial G}{\partial c}$ will also play an important rôle in the forthcoming development.

Similarly to (2), we approximate the state of our system as a pulled back spline

$$
\begin{equation*}
u_{h}=\sum_{i=1}^{N} u_{i} R_{i} \circ G^{-1} \tag{4}
\end{equation*}
$$

where $R_{i}^{g}, i=1, \ldots, N$ are B-splines. The expansion coefficients $u_{i}, i=1, \ldots, N$ will be found by solving a system of linear algebraic equations

$$
K_{c} \boldsymbol{u}=f_{c}
$$

As standard in the Galerkin approach, the elements of the stiffness matrix $K_{c}$ and the load vector $f_{c}$ are computed as $K_{c, i, j}=a_{G(] 0,1\left[^{d}\right)}\left(R_{j} \circ G^{-1}, R_{i} \circ G^{-1}\right)$, and $f_{c, i}=\ell_{G(] 0,1\left[^{d}\right)}\left(R_{i} \circ G^{-1}\right)$. Note that the dependence of $K_{\boldsymbol{c}}, f_{\boldsymbol{c}}$ on the control points is encapsulated in (2) and (4).

Already at this point the importance of geometry parametrization should be apparent. Indeed, at the very least it should be an invertible map, which is used to pull back the weak form of the PDE defined on the physical domain $\Omega$ into the parameter domain $] 0,1\left[{ }^{d}\right.$. In particular, for all $\boldsymbol{\xi} \in] 0,1\left[{ }^{d}\right.$ it is necessary that $\operatorname{det}(J(\boldsymbol{\xi}))>0 .{ }^{3}$ A sufficient condition, which guarantees the validity of the parametrization, is discussed in Appendix A.

[^7]
## 3 Boundary-driven approach to IGA shape optimization

In this section we will consider one possible approach to IGA shape optimization, which follows the ideas developed in [17, 9]. Within this framework we formulate the optimization problem in terms of boundary control points $\boldsymbol{c}^{t}$. The interior control points $\boldsymbol{c}^{i}$ for the geometry parametrization are treated as an implicit function of $\boldsymbol{c}^{t}$, see Appendix B. Additionally, the parametrization validity constraints $\operatorname{det}(J)>0$, or a sufficient condition for these (cf. Appendix A) have to be explicitly included into the problem formulation.

In order to compute domain parametrizations of high quality we rely upon minimizing the Winslow functional (Section B.2). However, to avoid solving a non-linear optimization problem at each shape optimization iteration, we construct a quadratic approximation to the Winslow functional around a reference parametrization, and update the reference parametrization when it becomes necessary to do so. Specifically, given a reference parametrization $G_{0}$ defined by the control points $\boldsymbol{c}_{0}$, to find a new parametrization we consider the quadratic programming problem

$$
\begin{equation*}
\min _{\Delta \boldsymbol{c}^{i}} \frac{1}{2} \Delta \boldsymbol{c}^{T} H\left(\boldsymbol{c}_{0}\right) \Delta \boldsymbol{c}+\nabla W\left(\boldsymbol{c}_{0}\right)^{T} \Delta \boldsymbol{c}+W\left(\boldsymbol{c}_{0}\right) \tag{5}
\end{equation*}
$$

where $W$ is the Winslow functional, and $H$ is its Hessian. The minimizer of this problem can be found by solving a linear system

$$
H\left(\boldsymbol{c}_{0}\right) \Delta \boldsymbol{c}=-\nabla W\left(\boldsymbol{c}_{0}\right)
$$

Using (3), this can be restated as

$$
\begin{equation*}
H_{c^{i}, c^{i}} \Delta \boldsymbol{c}^{i}=-\nabla W_{\boldsymbol{c}^{i}}-H_{c^{i}, c^{a}} \Delta \boldsymbol{c}^{t} \tag{6}
\end{equation*}
$$

where $\left[H_{c^{i}, c^{i}}\right]_{i j}=\partial^{2} W / \partial c^{i}{ }_{i} \partial c^{i}{ }_{j},\left[H_{c^{i}, c^{b}}\right]_{i j}=\partial^{2} W / \partial c^{i}{ }_{i} \partial c^{t}{ }_{j}$ and $\left[\nabla W_{c^{i}}\right]_{i}=\partial W / \partial c^{i}{ }_{i}$. The new parametrization is then defined by the control points given by $\boldsymbol{c}=\boldsymbol{c}_{0}+\Delta \boldsymbol{c}$.

With this in mind, to approximate (1) numerically we solve a sequence of subproblems

$$
\begin{array}{ll}
\max _{\Delta \boldsymbol{c}^{a}} & E(\boldsymbol{c}, \boldsymbol{u}) \\
\text { s.t. } & K_{\boldsymbol{c}} \boldsymbol{u}=f_{\boldsymbol{c}} \\
& H_{c^{i}, c^{i}} \Delta \boldsymbol{c}^{i}=-\nabla W_{\boldsymbol{c}^{i}}-H_{c^{i}, c^{a}} \Delta \boldsymbol{c}^{\star} \\
& \boldsymbol{c}=\boldsymbol{c}_{0}+\Delta \boldsymbol{c} \\
& \boldsymbol{d} \geq \varepsilon \\
& \boldsymbol{c}^{\star}{ }_{L} \leq \boldsymbol{c}^{\star} \leq \boldsymbol{c}^{\star}{ }_{U} \tag{7f}
\end{array}
$$

where $\boldsymbol{c}_{0}$ is the reference parametrization, $\Delta \boldsymbol{c}=\left(\Delta \boldsymbol{c}^{t}, \Delta \boldsymbol{c}^{i}\right)$, and (7e) is the sufficient condition for the validity of the parametrization discussed in Appendix A.

Reference parametrizations can be computed as follows. We minimize the Winslow functional as described in Appendix B, and check if the sufficient condition $\boldsymbol{d}>0$ is violated. If it is, then this condition is too strict and should be relaxed. To facilitate this we refine the spline space $\mathcal{S}_{\text {det }}$ where we compute expansion coefficients $\boldsymbol{d}$ of $\operatorname{det} J$. To reduce the number of constraints resulting from such refinement steps, we utilize local refinement. Specifically, we use Truncated Hierarchical B-splines (THB-splines) as basis functions. Note that it is important here to use the truncated version of hierarchical splines, since the partition of unity property (cf. [8]) implies that the spline control polygon converges locally to function values.

The refinement strategy we employ is as follows. For all indices $i$ that have negative spline expansion coefficient $d_{i} \leq 0$ of $\operatorname{det} J$, we refine the support of the associated basis function $R_{i}^{\text {det }}$. This is repeated until $d_{i}>0$ for all $i=1, \ldots, N_{\text {det }} .{ }^{4}$ In subproblem (7) we then put $\varepsilon=\rho \cdot \min _{i} d_{i}$ with $\rho=0.25$, see (11).

[^8]

Figure 1: Flowchart of the optimization algorithm.

The full optimization loop is illustrated in Figure 1. Note that we have chosen to carry out the spline space refinement described above only when the reference parametrization, and therefore also the subproblem (7), is updated. This allows us to keep the number of constraints constant when solving (7) numerically and therefore employ off-the-shelf optimization software. The initial guess for the non-linear Winslow optimization problem is generated using Coons' patches, see Appendix B.

## 4 Regularization-driven approach to IGA shape optimization

In this section we discuss an alternative approach to shape optimization using IGA, which does not involve explicit constraints on det $J$. The positions of the inner control points $\boldsymbol{c}^{i}$ enter this formulation as independent optimization variables, in the same way as $\boldsymbol{c}^{t}$. Consequently, we do not need to explicitly compute a domain parametrization, as this will be part of the outcome of the optimization process.

To this end we add Winslow functional $W$ as a regularization term to the objective function. Its role is to penalize configurations of control points that result in poor parametrizations. This idea has been used previously in the context of shape optimization in mechanics [6, 20]. Thus for a regularization parameter $\tau>0$ we consider the optimization problem

$$
\begin{array}{cl}
\min _{\boldsymbol{c}} & \tau W(\boldsymbol{c})-E(\boldsymbol{c}, \boldsymbol{u}), \\
\text { s.t. } & K_{\boldsymbol{c}} \boldsymbol{u}=f_{\boldsymbol{c}} \\
& \boldsymbol{c}_{L} \leq \boldsymbol{c} \leq \boldsymbol{c}_{U} \tag{8c}
\end{array}
$$

We put $W(\boldsymbol{c})=\infty$ if $\operatorname{det} J \leq 0$ at one of the quadrature points used for the integration when calculating $W(\boldsymbol{c})$. In this way the Winslow functional acts as a barrier term that ensures that


Figure 2: Sketch of the shape optimization problem. The goal is to find a shape of the reflector that maximizes the field close to a point.
the determinant is always positive at the quadrature points. This does not guarantee that it is positive everywhere, but it means that the numerics will not collapse due to a division by zero.

The regularization parameter $\tau$ needs to be tuned for the specific problem at hand. If it is too large the minimization will find a design with a small value of the Winslow functional but disregarding the objective $E(\boldsymbol{c}, \boldsymbol{u})$. If it is too small the optimization will find positions of the control points that have a low objective $E(\boldsymbol{c}, \boldsymbol{u})$, but with a poor parametrization, which might give a large discretization error of the discretized PDE. The appropriate values of $\tau$ would lead to a compromize between these two extreme situations. One simple strategy for choosing such a value is solve a sequence of problems (8) for decreasing values of $\tau .{ }^{5}$

## 5 Case study: Optimization of electromagnetic reflectors

In this section we will consider a 2D shape optimization problem originating from the field of electromagnetism. Our goal is to design a reflector that concentrates electrical energy in a desired area. This problem will serve as a model problem for comparing the two optimization approaches outlined in Sections 3 and 4.

### 5.1 Physical model

We consider a two dimensional scattering problem where a plane wave with frequency $f$ travels in a dielectric (air) and is scattered by two symmetric metallic (gold) reflectors, as depicted in Figure 2. Let $\epsilon_{c r}$ and $\mu_{r}$ denote the complex permittivity and permeability of the medium. Using the first order absorbing boundary condition [14] at the boundary $\Gamma_{t}$ of the truncated domain, the electromagnetic field $\hat{u}$ should satisfy the following PDE:

$$
\begin{align*}
\nabla \cdot\left(\frac{1}{\epsilon_{c r}} \nabla \hat{u}\right)+k_{0}^{2} \mu_{r} \hat{u}=0 & & \text { in } \Omega,  \tag{9a}\\
\frac{\partial\left(\hat{u}-u^{i}\right)}{\partial n}+\left(j k_{0}+\frac{1}{2 r_{t}}\right)\left(\hat{u}-u^{i}\right)=0 & & \text { on } \Gamma_{t} . \tag{9b}
\end{align*}
$$

[^9]| $f$ | $\mu_{r}$ | $\mu_{r}^{s}$ | $\sigma$ | $\epsilon_{0}$ | $\mu_{0}$ | $\epsilon_{r, \text { gold }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $4 \cdot 10^{14}[\mathrm{~Hz}]$ | 1.0 | 1.0 | $10^{6}[\mathrm{~S} / \mathrm{m}]$ | $\left(\mu_{0} c^{2}\right)^{-1}$ | $4 \pi 10^{-7}$ | $-20.199+j 1.381$ |

Table 1: Physical parameters

In the equations above, $k_{0}=2 \pi \sqrt{\epsilon_{0} \mu_{0}}$ is the wave number and $\epsilon_{0}, \mu_{0}$ refer to the permittivity and permeability of free space, respectively. The imaginary unit is denoted by $j$, the radius of the truncated domain is given by $r_{t}$, and $u^{i}$ is the incident plane wave, given by

$$
u^{i}(x, y)=e^{-j k_{0} \sqrt{\epsilon_{c r} \mu_{r}} x}
$$

The objective function of the shape optimization will be given by

$$
E(\boldsymbol{c}, u)=\int_{\Omega} \delta|\hat{u}|^{2} \mathrm{~d} x
$$

where $\delta$ is a Gaussian bell-function

$$
\delta(x, y)=e^{-\frac{x^{2}+y^{2}}{2 \alpha^{2}}}
$$

with $\alpha=0.1$. Thus we aim to focus the incoming energy in the vicinity of the origin $(0,0)$. The physical parameters that we use are given in Table 1. The complex permitivity of the reflector is calculated as $\epsilon_{c r}^{s}=\epsilon_{r, \text { gold }}-j \frac{\sigma}{\omega \epsilon_{0}}$.

The weak statement of the $\operatorname{PDE}(9)$ is to find $\hat{u} \in H^{1}(\Omega)$ such that for all test functions $\hat{v} \in H^{1}(\Omega)$ the following equality holds

$$
\begin{align*}
\int_{\Omega} \frac{1}{\epsilon_{c r}} \hat{\nabla} \hat{u} \cdot \hat{\nabla} \hat{v} \mathrm{~d} x+k_{0}^{2} \int_{\Omega} \mu_{r} \hat{u} \hat{v} \mathrm{~d} x+\left(j k_{0}+\frac{1}{2 r_{t}}\right) & \int_{\Gamma_{t}} \frac{1}{\epsilon_{c r}} \hat{u} \hat{v} \mathrm{~d} s \\
& =\frac{1}{\epsilon_{c r}} \int_{\Gamma_{t}}\left(\frac{\partial u^{i}}{\partial n}+\left(j k_{0}+\frac{1}{2 r_{t}}\right) u^{i}\right) \hat{v} \mathrm{~d} s \tag{10}
\end{align*}
$$

Due to the symmetry we only consider the upper half of the geometry shown in Figure 2. To accommodate the change of material parameters between the metallic reflector and the surrounding diellectric medium, we will split the domain into five patches, one for the reflector and the other four for the surrounding air. The layout is shown in Figure $3 .{ }^{6}$ Each patch is be parametrized using splines as described in Section 2. Using these parametrizations we can pull back the equation (10) to the parameter domain and apply the Galerkin method to it, which ultimately results in the system of linear algebraic equations, see Appendix C for details.

### 5.2 Results with boundary-driven approach

In this section we apply the method described in Section 3 to our model problem. We start with an initial design where the reflector has the shape of a circle. We consider two different spline spaces in which to approximate the PDE (9), namely using the knotvectors used for representing the geometry refined uniformly 3 and 4 times. Both spline spaces have degree $p=2$, and the number of degrees of freedoms are $N_{\text {coarse }}=2548$ and $N_{\text {fine }}=9300$, respectively. We will refer to these as the coarse and fine meshes.

We will use a tolerance tol $=10^{-3}$ when solving the subproblems (7) and a fixed number of reparametrizations, namely 10 when using the coarse mesh and 5 when using the fine mesh. We observed that using more reparametrizations did not lead to significant improvements in the

[^10]

Figure 3: Patch layout.
design. In our experiments the results with this method are sensitive to the number of quadrature points used when calculating the Winslow functional. To produce the results presented here we use 12 quadrature points per knot interval, to avoid under-integration ${ }^{7}$.

In our implementation we use the interior point solver from Ipopt to solve the subproblems (7). Doing this successfully required some parameter tuning. Namely, after solving one of the subproblems (7) some of the design bounds (7f) will be active; however, since Ipopt is an interior point algorithm, the starting point for the subsequent subproblem will be pushed away from the boundary as controlled by the parameter bound_push. We found that this parameter needs to be lower than the default value since the constraints on $\operatorname{det} J$ are quite sensitive and a relatively small perturbation of the control points might violate these constraints, which is undesirable. The default value is 0.1 , but in the experiments we set it to $10^{-5}$ instead.

Another key parameter is the barrier parameter mu_init. Specifically, we use the monotone strategy, where the barrier parameter is monotonically decreased as the optimization algorithm progresses. However, if this parameter is too large in the beginning of the algorithm we found that it will push the design towards configurations with large det $J$. To remedy this instead of the default value is 0.1 we use $10^{-4}$. For more information about the optimization algorithm implemented in Ipopt and it parameters see [22].

In Figure 4 the design at different stages of the optimization is presented, when using the coarse mesh. We observe that the design becomes increasingly hard to parametrize as the objective increases each time we change the reference parametrization. In Figure 5 the objective is plotted against the number of iterations. We see that the objective function increases after the reference parametrization is changed, but relatively quickly reaches a plateau. We already use a fairly large tolerance of $10^{-3}$ for the stopping criterion when solving the subproblems (7), however this behaviour indicates that it might help to relax the stopping criterion even further in these subproblems to improve the overall efficiency of the method. However to allow for a fair comparison between the two methods we do not investigate this further and use the same tolerance for both methods. The final objective, after 10 reparametrizations when using the coarse mesh, is $E_{h}=1.803$, however if we calculate the objective with a mesh that is refined uniformly twice we get $E_{h / 4}=1.556$, that is, a $16 \%$ difference. This means that there is actually a significant discretization error at this refinement level, which leads to an artificially large objective value.

As described in Section 3 we use local refinement to adapt the constraints on $\operatorname{det} J$ to the current design. This is done by using local refinements in the areas where the spline coefficients are non positive. The resulting meshes are plotted in Figure 6 to illustrate where the refinement is needed. We see that it is primarily inside the reflector and near the reflector-air interface, that this

[^11]

Figure 4: The designs at different stages of the optimization process, when using validity constraints and the coarse mesh. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure 5: The objective function during optimization process for the coarse and fine meshes. The vertical lines indicates where the parametrization is updated. $E_{h / 4}$ is the objective calculated on a refined mesh.


Figure 6: The mesh (knot lines), used for representing $\operatorname{det} J$ during the subproblems (7) for the constraints (11). The number of constraints is given by $N^{\text {det }}$ and the designs are obtained using the coarse mesh.


Figure 7: The final design using validity constraints and the fine mesh, $E_{h / 2}=1.638$
refinement is applied. We see that the number of constraints increases during the optimization process, but by no more than a factor of two. If we were to use uniform refinement the number of constraints would increase with more than a factor of 9 .

The final design, after 5 reparametrizations, when using the fine mesh is shown in Figure 7. We see that it is similar to the design obtained using the coarse mesh. The final objective here is $E_{h}=1.638$. After the mesh is refined uniformly the electrical energy is $E_{h / 4}=1.628$, which is only a $0.6 \%$ difference. The objective function during the optimization process is plotted in Figure 5. We see that the algorithm converges faster when using the fine mesh as no progress was observed after 5 reparametrizations.

### 5.3 Results with regularization-driven approach

In this section we will present the results obtained with the method described in Section 4. With this method we perform the optimization with all control points as optimization variables while using the Winslow functional as a regularization term. We will again use a tolerance of tol $=10^{-3}$ when solving the problem (8). The regularization parameter is set to $\tau=\frac{1}{8}$.

One can compare the design evolution shown in Figure 8 with those obtained previously, see Figure 4. The designs obtained using the regularization approach seem to have more regular parametrizations compared to those in Figure 4.

The final objective is $E_{h}=1.684$. If we calculate the electrical energy for this design on a twice refined mesh we get $E_{h / 4}=1.546$, that is, a difference of $9 \%$. This is less than the $16 \%$ we observed when using the boundary-driven method. This increase in accuracy might be due to the parametrization being of higher quality.

In Figure 9 we plot the electric energy $E_{h}$, the regularization term $\tau W$ and the objective function $\tau W-E_{h}$.

When using the method with the fine mesh we get the final design shown in Figure 10. We see that the shape of the reflector is very similar to that shown in Figure 8d. The main difference is that when using the fine mesh the parametrization is more regular, since the error in the discretization of the PDE is smaller, and therefore the optimization cannot exploit it to the same extend. This is especially notable at the bottom of the reflector were the inner control points where moved away from the point of interest when using the coarse mesh, as seen in Figure 8d. The final objective is $E_{h}=1.545$ and when evaluating it on a refined mesh we get the same result $E_{h / 4}=1.545$ with the difference at the 5th digit. The objective during the optimization process is plotted in Figure 9.

(a) The initial design, $E_{h}=0.202$

(c) 143 iterations, $E_{h}=1.593$

(b) 67 iterations, $E_{h}=1.461$

(d) The final design after 432 iterations , $E_{h}=1.683$

Figure 8: The designs at different stages of the optimization process when using the regularization approach. The reflector is outlined with a black line, and the control points that controls this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure 9: The objective function during the optimization process when using the regularization based approach, with the fine and coarse mesh.


Figure 10: The final design when using the regularization approach and the fine mesh, $E_{h / 2}=$ 1.545 .

The behaviour is similar for the two meshes, however the tolerance tol $=10^{-3}$ is reached with fewer iterations when using the fine mesh.

### 5.4 Comparison and discussion

In Table 2 we summarize the performance of the two methods. We report the objective after a fixed number of iterations, in this case after 100 iterations, the final objective computed on 3 different refinement levels and the average running time per iteration.

We observe that the average execution time per iteration is the same order of magnitude for the two methods. The main portion of the running times is spent on solving the state equation and computing the gradient of the objective function. The difference in the running time that we observed between the two methods might be due to a different number of function evaluations per iteration needed for trial steps of the algorithm. If we were to consider a larger problem, for example in 3D, the large number of validity constraints would likely lead to an increase in running time for the boundary-driven approach.

Regarding the quality of the designs we find, we note that shape optimization problems are prone to having many local optima, so it could be that the two methods find two different local optima. Therefore it can be futile to directly compare objective values. That being said, we observe that the boundary-driven approach happens to find solutions with slightly higher (better) objective value. On the other hand, the regularization based approach seems to estimate the objective value more accurately on coarser meshes, since our results were more reliable using this method, probably due to the better quality of the parametrizations it produced.

Also, observe that with the regularization based approach we only need the objective $E_{h}$, the Winslow functional and their first order derivatives. In addition we can solve a single optimization problem with design bounds as the only constraints. This means that the method is significantly easier to implement.

### 5.5 Conclusion

We described and compared two methods for shape optimization on spline-based representations. One uses validity constraints to enforce the validity of the geometry parametrization. The other

| Method | $E_{\text {iter }=100}$ | $E_{h}$ | $E_{h / 2}$ | $E_{h / 4}$ | $\frac{E_{h}-E_{h / 4}}{E_{h / 4}}$ | Avg time per iteration |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Linearizations coarse mesh | 1.482 | 1.803 | 1.589 | 1.556 | $16 \%$ | 9.68 sec |
| Regularization coarse mesh | 1.529 | 1.684 | 1.561 | 1.546 | $9 \%$ | 11.54 sec |
| Linearizations fine mesh | 1.495 | 1.638 | 1.629 | 1.628 | $0.6 \%$ | 53.5 sec |
| Regularization fine mesh | 1.482 | 1.545 | 1.545 | 1.545 | $0.003 \%$ | 71.0 sec |

Table 2: Comparison of the two methods from Section 3 and 4 on two different meshes. $E_{\text {iter }=100}$ is the objective after 100 iterations. $E_{h}$ is the objective computed on the mesh used in the optimization. $E_{h / 2}$ and $E_{h / 4}$ are the objectives computed after refining the mesh uniformly, once and twice. The execution time was measured on a 64 bit HP EliteBook 840 G4 with and Intel(R) Core(TM) i7-7500U CPU, with clock rate of 2.70 GHz .
uses a regularization term, and thus avoids both the validity constraints and the need of an explicit parametrization strategy altogether. We demonstrated how this simple approach performed comparably to the more complicated approach in terms of the final design, while requiring similar running times for the 2 D problem we considered. The regularization based approach seems to produce more reliable results and it is in addition much simpler to implement, since we only need the objective, the Winslow functional and their first order derivatives.

These results are encouraging and we plan to use the regularization based approach for 3D problems, where we expect that the efficiency advantages of the regularization approach will be more prominent, since the number of validity constraints for 3D parametrizations grows quickly.

## Funding

AE's research is funded by the Villum Fonden through the Villum Investigator Project InnoTop.

## References

[1] N. Aage, N. Mortensen, and O. Sigmund. Topology optimization of metallic devices for microwave applications. International Journal for numerical methods in engineering, 83(2):228248, 2010.
[2] R. E. Christiansen, J. Vester-Petersen, S. P. Madsen, and O. Sigmund. A non-linear material interpolation for design of metallic nano-particles using topology optimization. Computer Methods in Applied Mechanics and Engineering, 343:23-39, 2019.
[3] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. Isogeometric analysis: toward integration of $C A D$ and FEA. John Wiley \& Sons, 2009.
[4] A. Falini, J. peh, and B. Jttler. Planar domain parameterization with thb-splines. Computer Aided Geometric Design, 35-36:95 - 108, 2015. Geometric Modeling and Processing 2015.
[5] G. Farin and D. Hansford. Discrete coons patches. Computer aided geometric design, 16(7):691-700, 1999.
[6] J. Friederich, M. Scherer, and P. Steinmann. Isogeometric structural shape optimization using a fictitious energy regularization. PAMM, 11(1):709-710, 2011.
[7] P. Gangl, U. Langer, A. Mantzaflaris, and R. Schneckenleitner. Isogeometric simulation and shape optimization with applications to electrical machines. arXiv preprint arXiv:1809.03377, 2018.
[8] C. Giannelli, B. Jüttler, and H. Speleers. Thb-splines: The truncated basis for hierarchical splines. Computer Aided Geometric Design, 29(7):485-498, 2012.
[9] J. Gravesen, A. Evgrafov, D.-M. Nguyen, and P. Nørtoft. Planar parametrization in isogeometric analysis. In International Conference on Mathematical Methods for Curves and Surfaces, pages 189-212. Springer, 2012.
[10] P. C. Hansen. Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems. Numerical algorithms, 6(1):1-35, 1994.
[11] J. Hinz, A. Jaeschke, M. Möller, and C. Vuik. The role of pde-based parameterization techniques in gradient-based iga shape optimization applications. arXiv preprint arXiv:2001.10921, 2020.
[12] J. Hinz, M. Möller, and C. Vuik. Elliptic grid generation techniques in the framework of isogeometric analysis applications. Computer Aided Geometric Design, 65:48-75, 2018.
[13] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comp. Meth. Appl. Mech. Engrg., 194(39-41):41354195, 2005.
[14] J.-M. Jin. The finite element method in electromagnetics. Wiley, 1993.
[15] A. Mantzaflaris, B. Juettler, B. N. Khoromskij, and U. Langer. Low rank tensor methods in Galerkin-based isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 316:1062-1085, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
[16] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In Proceedings of the 2008 ACM symposium on Solid and physical modeling, pages 269-280, 2008.
[17] D. Nguyen, A. Evgrafov, and J. Gravesen. Isogeometric shape optimization for electromagnetic scattering problems. Progress in Electromagnetics Research B, 45:117-146, 2012.
[18] M. Pan and F. Chen. Low-rank parameterization of volumetric domains for isogeometric analysis. Computer-Aided Design, 114:82-90, 2019.
[19] S. Sajavičius, B. Jüttler, and J. Špeh. Template Mapping Using Adaptive Splines and Optimization of the Parameterization, pages 217-238. Springer International Publishing, Cham, 2019.
[20] M. Scherer, R. Denzer, and P. Steinmann. A fictitious energy approach for shape optimization. International Journal for Numerical Methods in Engineering, 82(3):269-302, 2010.
[21] A. Shamanskiy, M. H. Gfrerer, J. Hinz, and B. Simeon. Isogeometric parametrization inspired by large elastic deformation. arXiv preprint arXiv:1810.12425, 2018.
[22] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming, 106(1):25-57, 2006.
[23] E. Wadbro and C. Engström. Topology and shape optimization of plasmonic nano-antennas. Computer Methods in Applied Mechanics and Engineering, 293:155-169, 2015.
[24] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. Journal of computational physics, 1(2):149-172, 1966.
[25] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Optimal analysis-aware parameterization of computational domain in isogeometric analysis. In International Conference on Geometric Modeling and Processing, pages 236-254. Springer, 2010.
[26] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. Computer Methods in Applied Mechanics and Engineering, 200(23-24):2021-2031, 2011.
[27] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Constructing analysis-suitable parameterization of computational domain from cad boundary by variational harmonic method. Journal of Computational Physics, 252:275-289, 2013.
[28] Y. Zheng, M. Pan, and F. Chen. Boundary correspondence of planar domains for isogeometric analysis based on optimal mass transport. Computer-Aided Design, 114:28-36, 2019.

## A A sufficient condition for a valid parametrization

As pointwise constraints $\operatorname{det}(J)>0$ are generally speaking difficult to deal with, we utilize the spline nature of the geometry parametrization (2). Namely, when the geometry map $G \in C^{k}$ is a spline of degree $p$, then $\operatorname{det} J \in C^{k-1}$ is a spline with degree $d \cdot p-1$. This means that we can construct a spline space $\mathcal{S}_{\text {det }}$ that contains det $J .{ }^{8}$ We can therefore find the expansion coefficients $\boldsymbol{d}$ of $\operatorname{det} J$ with respect to B-splines in this space. ${ }^{9}$

Now we can use this expansion to derive a sufficient condition to replace the pointwise parametrization validity constraints $\operatorname{det}(J)>0$, by requiring that

$$
\begin{equation*}
\boldsymbol{d} \geq \varepsilon \tag{11}
\end{equation*}
$$

where $\varepsilon$ is a small positive algorithmic parameter. This condition guarantees that $\operatorname{det} J>0$ since B-splines are non-negative and form a partition of unity. This is not a necessary condition, meaning that we might have $d_{i} \leq 0$ for some $i$ even though $\left.\operatorname{det} J(\boldsymbol{\xi})>0, \forall \boldsymbol{\xi} \in\right] 0,1\left[{ }^{d}\right.$. But if the spline space $\mathcal{S}_{\text {det }}$ is refined then the spline expansion coefficients will move closer to values of the spline. So if $\varepsilon$ is small enough and $\operatorname{det} J>0$ then the constraint is likely to be satisfied for a sufficiently refined spline space $\mathcal{S}_{\text {det }}$.

## B Domain parameterization techniques

In this section we review some techniques of finding a parametrization of the interior given the boundary. In IGA this comes down to finding the position of the inner control points given boundary control points. We do not aim to give a thorough review of all the techniques that are available, as this is out of the scope of this work. We will only introduce the methods that are related, or directly used, by the two shape optimization methods considered in this work

[^12]
## B. 1 Coons patch and spring method

Two simple methods for constructing a grid of control points is the Coons patch [5] and the spring method [9]. They both produce inner control points that depend linearly on the boundary control points, however they only produce valid parametrizations for geometries that are not too complicated. In this work we will use these methods for finding an initial guess for the optimization based approach that is described below.

## B. 2 Optimization-based techniques

A more complex, and in general nonlinear, class of parametrization methods consists of optimizationbased methods. Here the geometry map $G$ is chosen such that it minimizes a quality metric $w(\boldsymbol{\xi})$

$$
\begin{align*}
& \min _{G} \int_{] 0,1\left[^{d}\right.} w(\boldsymbol{\xi}) \mathrm{d} \boldsymbol{\xi},  \tag{12}\\
& \text { s.t. }\left.G\right|_{\partial[0,1]^{d}}=\gamma, \tag{13}
\end{align*}
$$

where $\gamma$ is a given boundary curve.
There are several different quality metrics to choose from, cf. [9, 27]. In this work we will consider the Winslow functional which is given in terms of the Jacobian matrix $J$ as

$$
\begin{equation*}
W=\int_{10,1\left[^{d}\right.} w(\boldsymbol{\xi}) \mathrm{d} \boldsymbol{\xi} \tag{14}
\end{equation*}
$$

with

$$
\begin{equation*}
w=\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J} \tag{15}
\end{equation*}
$$

In 2D the Winslow functional has the nice property that its minimizer has a harmonic inverse [9]. This guarantees that the minimizer of (12) is unique and bijective, i.e. det $J \neq 0$. It should be noted that this minimizer is not necessarily a spline, so looking for a spline parametrization on the form (2) by minimizing (12) with the Winslow functional will only guarantee a valid parametrization if the spline space used for the parametrization has high enough resolution.

Within the shape optimization context we also need first and second order partial derivatives of $w$. The derivative with respect to a parameter $\alpha$ is given by

$$
\begin{equation*}
\frac{\partial w}{\partial \alpha}=2(\operatorname{det} J)^{-1} \operatorname{tr}\left(J^{T} \frac{\partial J}{\partial \alpha}\right)-\operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \alpha}\right) \frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J} \tag{16}
\end{equation*}
$$

where we used the relation $\frac{\partial}{\partial \alpha} \operatorname{det} J=\operatorname{det} J \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \alpha}\right)$. The second order derivative is given by

$$
\begin{align*}
\frac{\partial^{2} w}{\partial \alpha \partial \beta} & =2(\operatorname{det} J)^{-1} \operatorname{tr}\left(\frac{\partial J^{T}}{\partial \alpha} \frac{\partial J}{\partial \beta}\right) \\
& -2(\operatorname{det} J)^{-1} \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \alpha}\right) \operatorname{tr}\left(J^{T} \frac{\partial J}{\partial \beta}\right) \\
& -2(\operatorname{det} J)^{-1} \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \beta}\right) \operatorname{tr}\left(J^{T} \frac{\partial J}{\partial \alpha}\right)  \tag{17}\\
& +\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J} \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \alpha}\right) \operatorname{tr}\left(J^{-1} \frac{\partial J}{\partial \beta}\right) \\
& +\frac{\operatorname{tr}\left(J^{T} J\right)}{\operatorname{det} J} \operatorname{tr}\left(\frac{\partial J^{T}}{\partial \beta} J^{-1} \frac{\partial J}{\partial \alpha} J^{-1}\right)
\end{align*}
$$

using the fact the fact that $\frac{\partial}{\partial \alpha} J^{-1}=-J^{-1} \frac{\partial J}{\partial \alpha} J^{-1}$ and assuming that $J$ depends linearly on $\alpha$, which is the case when $\alpha$ is a coordinate of a control point. Calculation of $\frac{\partial w}{\partial \alpha}$ and $\frac{\partial^{2} w}{\partial \alpha \partial \beta}$ can be implemented as an assembly of a linear and bilinear forms within IGA framework. In G+Smo this can for instance be accomplished using the gsExprEvaluator class that is typically employed for isogeometric stiffness matrix assembly purposes.

## C IGA discretization details

Using the spline parametrizations of patch geometries we pull back the weak form (10) to the parameter domain, which results in the following equation:

$$
\begin{align*}
\int_{] 0,1\left[^{2}\right.} \frac{1}{\epsilon_{c r}} J^{-T} \nabla u \cdot J^{-T} \nabla v & |\operatorname{det} J| \mathrm{d} \boldsymbol{\xi}+k_{0}^{2} \int_{] 0,1\left[^{2}\right.} \mu_{r} u v|\operatorname{det} J| \mathrm{d} \boldsymbol{\xi} \\
& +\left(j k_{0}+\frac{1}{2 r_{t}}\right) \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}} u v\left|\frac{\partial G}{\partial t}\right| \mathrm{d} t \\
& =\frac{1}{\epsilon_{c r}} \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}}\left(\frac{\partial u^{i}}{\partial n} \circ G+\left(j k_{0}+\frac{1}{2 r_{t}}\right) u^{i} \circ G\right) v\left|\frac{\partial G}{\partial t}\right| \mathrm{d} t \tag{18}
\end{align*}
$$

where we have $u=\hat{u} \circ G, v=\hat{v} \circ G$ and where $t$ is the parameter on the boundary. After applying the Galerkin method to (18), we arrive at the linear system of linear algebraic equations

$$
\begin{equation*}
A \boldsymbol{u}=(K+M+T) \boldsymbol{u}=\boldsymbol{f} \tag{19}
\end{equation*}
$$

where $\boldsymbol{u}=\left(u_{1}, \ldots, u_{N}\right)^{T}$ and $K, M, T$ and $f$ is given by

$$
\begin{align*}
K_{k l} & =\int_{[0,1]^{2}} \frac{1}{\epsilon_{c r}} J^{-T} \nabla R_{k} \cdot J^{-T} \nabla R_{l}|\operatorname{det} J| \mathrm{d} \boldsymbol{\xi}  \tag{20a}\\
M_{k l} & =k_{0}^{2} \int_{[0,1]^{2}} \mu_{r} R_{k} R_{l}|\operatorname{det} J| \mathrm{d} \boldsymbol{\xi}  \tag{20b}\\
T_{k l} & =\left(j k_{0}+\frac{1}{2 r_{t}}\right) \int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}} R_{k} R_{l}\left|\frac{\partial G}{\partial t}\right| \mathrm{d} t  \tag{20c}\\
f_{l} & =\int_{G^{-1}\left(\Gamma_{t}\right)} \frac{1}{\epsilon_{c r}}\left(\frac{\partial u^{i}}{\partial n} \circ G+\left(j k_{0}+\frac{1}{2 r_{t}}\right) u^{i} \circ G\right) e_{l}\left|\frac{\partial G}{\partial t}\right| \mathrm{d} \xi \tag{20d}
\end{align*}
$$

Note that the values of $\varepsilon_{c r}$ and $\mu_{r}$ are set to the properties of gold in patch 4 and for air in the other patches. Owing to the restriction of the IGA library we are utilizing, we further reformulate the system of complex algebraic equations (19) as

$$
\left[\begin{array}{cc}
\Re(A) & -\Im(A) \\
-\Im(A) & -\Re(A)
\end{array}\right]\left[\begin{array}{c}
\Re(\boldsymbol{u}) \\
\Im(\boldsymbol{u})
\end{array}\right]=\left[\begin{array}{c}
\Re(\boldsymbol{f}) \\
-\Im(\boldsymbol{f})
\end{array}\right]
$$

## APPENDIX <br> B

## Results using Regularizations

In this appendix we give some results from the optimization problem described in Section 3.3, for additional choices of the regularization parameter $\tau$.

(a) $\tau=\frac{1}{4}, E_{h}=0.973, E_{h / 4}=0.980,165$ iterations

(c) $\tau=\frac{1}{16}, E_{h}=1.645, E_{h / 4}=1.646,357$ iterations

(b) $\tau=\frac{1}{8}, E_{h}=1.373, E_{h / 4}=1.377,310$ iterations

(d) $\tau=\frac{1}{32}, E_{h}=2.227, E_{h / 4}=1.457$, the optimization algorithm terminated from reaching the maximum number of iterations (3000)

Figure B.1: The final designs when using the regularization based approach and the coarse mesh for different values of the regularization parameter $\tau$. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.


Figure B.2: The optimization history when using the regularization based method and the coarse mesh for different values of $\tau$. We plot the electrical energy $E_{h}$, the regularization term and $E_{h}-\tau W$ which is the actual function that is maximized. ${ }^{1}$ For $\tau=1 / 32$ the optimization algorithm terminated from reaching the maximum number of iterations.

(a) $\tau=\frac{1}{4}, E_{h}=0.982, E_{h / 4}=0.982,199$ iterations

(c) $\tau=\frac{1}{16}, E_{h}=1.646, E_{h / 4}=1.646,382$ iterations

(e) $\tau=\frac{1}{64}, E_{h}=1.688, E_{h / 4}=1.688,581$
iterations iterations

(b) $\tau=\frac{1}{8}, E_{h}=1.376, E_{h / 4}=1.376,395$ iterations

(d) $\tau=\frac{1}{32}, E_{h}=1.680, E_{h / 4}=1.680,443$ iterations

(f) $\tau=\frac{1}{128}, E_{h}=1.734, E_{h / 4}=1.688,734$ iterations

Figure B.3: The final designs when using the regularization based approach and the fine mesh for different values of the regularization parameter $\tau$.


Figure B.4: The optimization history when using the regularization based method and the fine mesh for different values of $\tau$. We plot the electrical energy $E_{h}$, the regularization term and $E_{h}-\tau W$ which is the actual function that is maximized.


Figure B. 5
Figure B.6: Results for different starting guesses, when using the fine mesh. In the top row we show the starting guesses, in the middle row the final design when using the regularization based approach. The reflector is outlined with a black line, and the control points of this boundary is colored black. The grey lines are parameter lines mapped with the geometry map, to illustrate the parametrization.

## APPENDIX C

## Perfectly Matched

## Layers

In this appendix we will explain how to use PML for free surface flow. We will start by introducing the method in 1D and then extend this result to the 3D PDE of interest. We will consider the Helmholtz equation in 1D

$$
\begin{equation*}
-\ddot{\phi}-K \phi=0 \quad \text { in } \mathbb{R} \tag{C.1}
\end{equation*}
$$

The method was first proposed in [5] with the idea of replacing absorbing boundary conditions with absorbing layers. The formulation was in the beginning derived by computing solutions for incident waves and solving for the conditions where the reflections vanish. However it was later found that PML formulations can be derived in a more elegant and general way, by using a complex-coordinate stretch [8, 40]. This technique is what we shall consider here. The main idea is as follows:

The solution to (C.1) is a plane wave

$$
h(x)=\mathrm{e}^{-\mathrm{j} K x} .
$$

This solution is analytic at can therefore be extended to $\mathbb{C}$ by analytic continuation [10]. Now if we consider this solution along a trajectory that goes into the complex plane $\gamma(s)=s+\mathrm{j} f(s)$ then we get

$$
\begin{equation*}
h(\gamma(s))=\mathrm{e}^{-\mathrm{j} K x} \mathrm{e}^{K f(s)} \tag{C.2}
\end{equation*}
$$

which decays exponentially when $f(s)<0$.
Now we choose a trajectory such that is on the real line within a domain of interest $[-l, l]$ and then bends into the complex plane on the intervals $] l, L]$ and $[-L, l[$. The solution along this trajectory will be the same as the original solution within the domain of interest $[0, l]$, and the radiating waves will decay in the layers $] l, L]$ and $[-L, l[$. Inspired by $[3]$ we will here consider trajectories on the form

$$
\gamma(s)= \begin{cases}s & \text { if }-l<s<l,  \tag{C.3}\\ s+\mathrm{j} \frac{C}{\omega}\left(\frac{-s-l}{L-l}\right)^{n} & \text { if }-L<s<-l, \\ s-\mathrm{j} \frac{C}{\omega}\left(\frac{s-l}{L-l}\right)^{n} & \text { if } l<s<L,\end{cases}
$$

with parameters $L, l, n, C \in \mathbb{R}$. An example of such a trajectory is shown in Figure C.1. The function $\gamma$ is called the stretching function.


Figure C.1: The streching function $\gamma$ for $l=2, L=3, n=2, C=5$ and for $\omega=\sqrt{K \cdot G}$ with gravitational constant $g=9.82$ and wavenumber $K=3$.

We denote complex derivatives as ' or $\frac{\partial}{\partial z}$ and let ${ }^{\text {' }}$ or $\frac{\partial}{\partial x}$ denote real derivatives in the physical space.

We will now solve for the analytic extension along the trajectory $\gamma$, so we will consider the equation

$$
\begin{array}{rlr}
-\phi^{\prime \prime}-K \phi=0 & & \text { in } \gamma([-L, L]) \\
\phi^{\prime} & =0 &  \tag{C.4b}\\
\text { on } \partial \gamma([-L, L])
\end{array}
$$

where $[-L, L]$ is the truncated domain. As the wave decays exponentially in the PML region, it is tiny when it reaches the boundary of the PML region, so it is not so important which boundary condition we use here [26]. In this work we shall use homogenous Neumann conditions, $\phi^{\prime}=0$ at $z=\gamma(-L)$ and $z=\gamma(L)$. We can now multiply with a test function $\psi \in H^{1}(\gamma([-L, L]))$ to get

$$
\begin{equation*}
-\int_{\gamma([-L, L])} \phi^{\prime \prime} \psi \mathrm{d} z-K \int_{\gamma([-L, L])} \phi \psi \mathrm{d} z=0 \tag{C.5}
\end{equation*}
$$

Note here that $\mathrm{d} z$ denotes complex integration.
If the functions $\gamma$ and $\phi$ are holomorphic, i.e., complex valued functions that are complex differentiable in every point, then we have that

$$
\frac{\partial}{\partial x}\left(\phi_{S} \circ \gamma\right)=\frac{\mathrm{d} \gamma}{\mathrm{~d} x} \frac{\partial \phi_{S}}{\partial z} \circ \gamma=\dot{\gamma} \frac{\partial \phi_{S}}{\partial z} \circ \gamma
$$

so along $\gamma$ we have that

$$
\frac{\partial}{\partial z}=\frac{1}{\dot{\gamma}} \frac{\partial}{\partial x}
$$

and

$$
\mathrm{d} z=\dot{\gamma} \mathrm{d} x
$$

We can use this to pull pack the equation (C.5) with $\gamma$. This gives

$$
\begin{equation*}
-\int_{-L}^{L} \frac{1}{\dot{\gamma}} \frac{\mathrm{~d}}{\mathrm{~d} x}\left(\frac{1}{\dot{\gamma}} \phi^{\prime}\right) \psi \dot{\gamma} \mathrm{d} x-K \int_{-L}^{L} \phi \psi \dot{\gamma} \mathrm{~d} x=0 \tag{C.6}
\end{equation*}
$$

We can now use Greens identity to get

$$
\begin{equation*}
-\int_{-L}^{L} \frac{1}{\dot{\gamma}} \frac{\mathrm{~d} \phi}{\mathrm{~d} x} \frac{1}{\dot{\gamma}} \frac{\mathrm{~d} \psi}{\mathrm{~d} x} \dot{\gamma} \mathrm{~d} x=0 \tag{C.7}
\end{equation*}
$$

If we parametrize the interval $[-L, L]$ with splines, we can pull back (C.7) to the parameter domain, and derive the system matrices with the same approach as demonstrated in Section 1.2.

We will now consider how to use the method for the 3D problem

$$
\begin{array}{rlrl}
-\widetilde{\Delta} \phi & =0 & \text { in } \Omega_{\infty} \\
K \phi-\frac{\partial \phi}{\partial z} & =0 & \text { on } \Gamma_{f}, \\
\frac{\partial \phi}{\partial \boldsymbol{n}} & =0 & \text { on } \Gamma_{\text {Symm }}, \\
\frac{\partial \phi}{\partial \boldsymbol{n}} & =0 & & \text { on } \Gamma_{s}, \tag{C.8d}
\end{array}
$$

for $\phi: \mathbb{R}^{3} \rightarrow \mathbb{C}$. Here $\widetilde{\Delta}$ is the laplace operator in physical space and $\Gamma_{f}$ is the free surface $z=0$.

We will let $x_{1}, x_{2}, x_{3}$ denote the 3 components of $\boldsymbol{x} \in \mathbb{R}^{3}$ as we will use $z_{1}, z_{2}, z_{3}$ to denote the componenents of $\boldsymbol{z} \in \mathbb{C}^{3}$.

The trajectory given in (C.3) is univariate, so to use the technique for our 3D domain we will consider uniaxial stretching. That is for each direction $i=1,2,3$ we will use a stretching function $\gamma_{i}$ on the form (C.3) with parameters $L_{i}, l_{i}$. Note that in the z-direction of our problem the field already decays exponentially, so in principle PML is not needed in this direction, as long as our domain is deep enough. There exists techniques to speed up this decay via real stretching [26]. However as this work is focused on the application of IGA and not PML we shall not investigate this further, and we will apply the stretching function above in all directions.

If we let $J_{\gamma}$ denote the Jacobian of the map $\left(x_{1}, x_{2}, x_{3}\right)^{T} \mapsto\left(\gamma_{1}\left(x_{1}\right), \gamma_{2}\left(x_{2}\right), \gamma_{3}\left(x_{3}\right)\right)^{T}$, we can write the weak formulation as: Find $\phi \in H^{1}(\Omega)$ such that

$$
\begin{equation*}
\int_{\Omega}\left\langle J_{\gamma}^{-1} \widetilde{\nabla} \phi_{S}, J_{\gamma}^{-1} \widetilde{\nabla} \psi\right\rangle \operatorname{det} J_{\gamma} \mathrm{d} V+K \int_{\Gamma_{f}} \phi_{S} \psi \operatorname{det} J_{\gamma} \mathrm{d} A=-\int_{\Gamma_{s}} \frac{\partial \phi_{I}}{\partial \boldsymbol{n}} \psi \mathrm{~d} A \tag{C.9}
\end{equation*}
$$

for all test functions $\psi \in H^{1}(\Omega)$. Where we have used that $\Gamma_{s}$ is within the domain of interest $\Omega_{0}$ where $\gamma_{i}(s)=s$ for all $i=1,2,3$. Note that it is sufficient for this to hold for real test functions since (C.9) is linear in $\psi$.

We will now pull back this weak formulation to the parameter domain yielding.

$$
\begin{align*}
& \int_{\Omega}\left\langle J_{\gamma}^{-1} J^{-T} \nabla u, J_{\gamma}^{-1} J^{-T} \nabla v\right\rangle \operatorname{det} J_{\gamma} \operatorname{det} J \mathrm{~d} V+ \\
& K \int_{\Gamma_{f}} u v \operatorname{det} J_{\gamma}\left\|\frac{\partial G}{\partial \nu_{1}} \times \frac{\partial G}{\partial \nu_{2}}\right\| \mathrm{d} A=-\int_{\Gamma_{s}} \frac{\partial \phi_{I}}{\partial \boldsymbol{n}} \circ G v \mathrm{~d} A \tag{C.10}
\end{align*}
$$

where $u=\phi_{S} \circ G$ and $v=\psi \circ G$.
We will in this work use the parameters $l_{i}=3$ and $L_{i}=5$ for $i=1,2$ and $l_{3}=1$ and $L_{3}=1, C=5$ and $n=2$.

## Bibliography

[1] N. Aage, N. Mortensen, and O. Sigmund. "Topology optimization of metallic devices for microwave applications". In: International Journal for numerical methods in engineering 83.2 (2010), pages 228-248.
[2] G. Allaire, F. Jouve, and A.-M. Toader. "A level-set method for shape optimization". In: Comptes Rendus Mathematique 334.12 (2002), pages 1125-1130.
[3] A. V. Astaneh, B. Keith, and L. Demkowicz. "On perfectly matched layers for discontinuous Petrov-Galerkin methods". In: Computational Mechanics 63.6 (2019), pages 1131-1145.
[4] M. P. Bendsøe and O. Sigmund. Optimization of structural topology, shape, and material. Volume 414. Springer, 1995.
[5] J.-P. Berenger et al. "A perfectly matched layer for the absorption of electromagnetic waves". In: Journal of computational physics 114.2 (1994), pages 185200.
[6] H. Brass and K. Petras. Quadrature theory: the theory of numerical integration on a compact interval. 178. American Mathematical Soc., 2011.
[7] L. Chen et al. "Structural shape optimization of three dimensional acoustic problems with isogeometric boundary element methods". In: Computer Methods in Applied Mechanics and Engineering 355 (2019), pages 926-951.
[8] W. C. Chew and W. H. Weedon. "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates". In: Microwave and optical technology letters 7.13 (1994), pages 599-604.
[9] E. Cohen and L. L. Schumaker. "Rates of convergence of control polygons". In: Computer Aided Geometric Design 2.1-3 (1985), pages 229-235.
[10] E. T. Copson. An introduction to the theory of functions of a complex variable. Volume 158. Oxford London, 1935.
[11] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. Isogeometric analysis: toward integration of CAD and FEA. John Wiley \& Sons, 2009.
[12] W. J. Duncan. Galerkin's Method in Mechanics and Differential Equations. Technical report. Aeronautical Research Council London (England), 1937.
[13] G Farin. "Konstruktion und Eigenschaften von Bézier-Kurven und Bézier-Flächen". In: Diplom-Arbeit. Technische Universität Braunschweig, 1977.
[14] G. Farin and D. Hansford. "Discrete coons patches". In: Computer aided geometric design 16.7 (1999), pages 691-700.
[15] J. Friederich, M. Scherer, and P. Steinmann. "Isogeometric structural shape optimization using a fictitious energy regularization". In: PAMM 11.1 (2011), pages 709-710.
[16] W. Gander and W. Gautschi. "Adaptive quadrature - revisited". In: BIT Numerical Mathematics 40.1 (2000), pages 84-101.
[17] P. Gangl et al. "Isogeometric simulation and shape optimization with applications to electrical machines". In: arXiv preprint arXiv:1809.03377 (2018).
[18] T. Gerstner and M. Griebel. "Dimension-adaptive tensor-product quadrature". In: Computing 71.1 (2003), pages 65-87.
[19] C. Giannelli, B. Jüttler, and H. Speleers. "THB-splines: The truncated basis for hierarchical splines". In: Computer Aided Geometric Design 29.7 (2012), pages 485-498.
[20] J. Gravesen et al. "Planar parametrization in isogeometric analysis". In: International Conference on Mathematical Methods for Curves and Surfaces. Springer. 2012, pages 189-212.
[21] J. Harrison. "Fast and accurate Bessel function computation". In: 2009 19th IEEE Symposium on Computer Arithmetic. IEEE. 2009, pages 104-113.
[22] J Hinz, M. Möller, and C. Vuik. "Elliptic grid generation techniques in the framework of isogeometric analysis applications". In: Computer Aided Geometric Design 65 (2018), pages 48-75.
[23] J. Hinz et al. "The Role of PDE-Based Parameterization Techniques in GradientBased IGA Shape Optimization Applications". In: arXiv preprint arXiv:2001.10921 (2020).
[24] T. Hirschler et al. "Isogeometric sizing and shape optimization of thin structures with a solid-shell approach". In: Structural and Multidisciplinary Optimization 59.3 (2019), pages 767-785.
[25] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: Computer methods in applied mechanics and engineering 194.39-41 (2005), pages 41354195.
[26] S. G. Johnson. Notes on Perfectly Matched Layers (PMLs). 2010.
[27] S. K. Khattri. "Grid generation and adaptation by functionals". In: Computational E Applied Mathematics 26.2 (2007), pages 235-249.
[28] H. Lian, P Kerfriden, and S. Bordas. "Shape optimization directly from CAD: An isogeometric boundary element approach using T-splines". In: Computer Methods in Applied Mechanics and Engineering 317 (2017), pages 1-41.
[29] N. D. Manh et al. "Isogeometric shape optimization of vibrating membranes". In: Computer Methods in Applied Mechanics and Engineering 200.13-16 (2011), pages 1343-1353.
[30] A. Mantzaflaris. Geometry plus Simulation Modules (G+Smo). 2018.
[31] J. Newman. "Cloaking a circular cylinder in water waves". In: European journal of mechanics-B/fluids 47 (2014), pages 145-150.
[32] J. N. Newman. Marine hydrodynamics. The MIT press, 2018.
[33] D. Nguyen, A. Evgrafov, and J. Gravesen. "Isogeometric Shape Optimization for Electromagnetic Scattering Problems". English. In: Progress in Electromagnetics Research B 45 (2012), pages 117-146. ISSN: 1937-6472.
[34] D. M. Nguyen. "Isogeometric analysis and shape optimization in electromagnetism". In: (2012).
[35] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/, Release 1.0.27 of 2020-06-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. URL: http://dlmf.nist.gov/.
[36] P. Nørtoft and J. Gravesen. "Isogeometric shape optimization in fluid mechanics". In: Structural and Multidisciplinary Optimization 48.5 (2013), pages 909925.
[37] M. Scherer, R. Denzer, and P. Steinmann. "A fictitious energy approach for shape optimization". In: International Journal for Numerical Methods in Engineering 82.3 (2010), pages 269-302.
[38] U. Schwarzmaira. "Isogeometric fluid analysis with applications in turbine design". PhD thesis. Johannes Kepler University Linz, November 2012.
[39] A. Shamanskiy et al. "Isogeometric parametrization inspired by large elastic deformation". In: Computer Methods in Applied Mechanics and Engineering 363 (2020), page 112920.
[40] F. L. Teixeira and W. C. Chew. "General closed-form PML constitutive tensors to match arbitrary bianisotropic and dispersive linear media". In: IEEE Microwave and Guided Wave Letters 8.6 (1998), pages 223-225.
[41] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: Mathematical programming 106.1 (2006), pages 25-57.
[42] A. M. Winslow. "Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh". In: Journal of computational physics 1.2 (1966), pages 149-172.
[43] G. Xu et al. "Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method". In: Journal of Computational Physics 252 (2013), pages 275-289.
[44] G. Xu et al. "Optimal analysis-aware parameterization of computational domain in isogeometric analysis". In: International Conference on Geometric Modeling and Processing. Springer. 2010, pages 236-254.
[45] G. Xu et al. "Parameterization of computational domain in isogeometric analysis: methods and comparison". In: Computer Methods in Applied Mechanics and Engineering 200.23-24 (2011), pages 2021-2031.
[46] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. The finite element method: its basis and fundamentals. Elsevier, 2005.


[^0]:    General rights
    Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

    - Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
    - You may not further distribute the material or use it for any profit-making activity or commercial gain
    - You may freely distribute the URL identifying the publication in the public portal

    If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

[^1]:    ${ }^{1}$ https://github.com/gismo
    ${ }^{2}$ https://github.com/coin-or/ipopt

[^2]:    ${ }^{1}$ This approach is from private conversations with Jochen Hinz

[^3]:    ${ }^{1}$ In practice we set a limit of the number of refinements, and return an error if this limit is reached.

[^4]:    ${ }^{2}$ Note that since we are not using NURBS, the shape of truncated domain is in fact an approximation to the half circle and therefore both coordinates of the associated control points has to be fixed. If we were to use NURBS here you could in principle also allow these to move while preserving the shape of the truncated domain.

[^5]:    ${ }^{1}$ The domain was suggested and supplied by Angelos Mantzaflaris and it originates from [38].

[^6]:    ${ }^{1}$ https://github.com/gismo
    2https://github.com/coin-or/Ipopt

[^7]:    ${ }^{3}$ It is equivalent to require that $\operatorname{det}(J(\boldsymbol{\xi}))<0$, however in this work we will use the constraint $\operatorname{det}(J(\boldsymbol{\xi}))>0$

[^8]:    ${ }^{4}$ In practice we terminate this procedure either if $\boldsymbol{d}>0$ or when a maximum level of refinement ( 7 in our numerical experiments) is attained. The latter termination criterion has not been observed in our experiments.

[^9]:    ${ }^{5} \mathrm{We}$ should note that the literature on regularization is quite extensive, see for example [10] and references therein, and this topic is somewhat beyond the scope of this work.

[^10]:    ${ }^{6}$ For automatic generation of patch layouts the interested reader is referred to [28].

[^11]:    ${ }^{7}$ In the IGA formulation (20) we also integrate non-polynomials. We use 3 quadrature points for the mass matrix $M$ and 7 quadrature points for the stiffness matrix $K$.

[^12]:    ${ }^{8}$ Specifically, the smallest spline space containing $\operatorname{det} J$ can be obtained by increasing the multiplicity of each knot in geometry map knotvector by $(d-1) \cdot p$ to account for the reduction in the differentiability and increase of the degree.
    ${ }^{9}$ This could be done using either interpolation or $L^{2}$ projection. In either approach one has to solve a linear system to obtain the expansion coefficients. The linear system matrix needs to be inverted only once for a given spline basis and, moreover, the matrix has a Kronecker/separable structure (cf. [15]) therefore the solution can be obtained extremely fast.

