Practical isogeometric shape optimization: Parameterization by means of regularization

A Limkilde, A Evgrafov, J Gravesen, Angelos Mantzaflaris

Practical isogeometric shape optimization: Parameterization by means of regularization

A. Limkilde[1,*], A. Evgrafov[2], J. Gravesen[1], and A. Mantzaflaris[4]

[1] *Department of Applied Mathematics and Computer Science, Technical University of Denmark*
[2] *Department of Mathematical Sciences, Aalborg University*
[4] *Inria Sophia Antipolis, Université Côte d'Azur*
[*] *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 intractably 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 [22] is a Galerkin method that uses splines to approximate both the geometric domain and the 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 [9]. 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 [42, 20, 17, 32]. 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 [17, 46] 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 which has harmonic inverse. In [20] this property is reformulated as a nonlinear PDE and the parametrization is found by solving this PDE. In [19] this PDE based parametrization technique is used for a gradient based shape optimization algorithm with IGA. In [17, 33] the same property is attained by minimizing the Winslow functional [45]. The method can be made more flexible by the use of adaptive splines [11], that allow to enrich the feasible region near to complex boundaries. In the recent works [42, 39] the approach of parametrizing a complex domain by deforming a given template is explored. In [35] 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 [14, 33]. These can be enforced either by using injectivity cones or by using the spline coefficients of the Jacobian determinant [47]. 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 an 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 define 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 a design with a valid parametrization. Such an approach has been considered in the context of shape optimization in mechanics in [40], and has to the best of our knowledge, only been considered very briefly in the context of shape optimization with IGA in [13]. 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.

Different frameworks for shape optimization with IGA exist in the literature. For example in [29] a tetrahedral mesh is used to represent the interior of the computational domain, while still representing the boundary using splines. The authors rely on external mesh generator software, while we in this work aim to avoid mesh generation by instead maintaining a parametrization of the interior. In [15] the shape is represented implicitly as the level set of a function. The PDE is then posed on a design domain, which remains constant during the optimization, and the level set function enters the formulation by the ersatz material approach. This framework has the advantage that it allows for changes in the topology of the shape, however at the same time the final shape is represented as a level set and therefore post processing is required to represent this shape using splines, which is necessary for importing the shape into CAD software. In our work the optimization is performed directly on the spline representation and thus the result can be readily imported into CAD software after the optimization. Another possible approach is to use the isogeometric representation of geometry combined with the boundary element method (IGABEM) [28, 6]. In IGABEM the PDE is reduced to an integral formulation on the boundary of the domain, and therefore maintaining a parametrization of the interior of the domain is avoided altogether. However in this case one has to deal with the standard complications of BEM, in particular the fact that the system matrices are dense, non-symmetric, and costly to compute. Additionally the Greens function for the considered PDE has to be known, which is not always the case.

In this work we illustrate that the simple regularization based approach is able to handle complicated geometries, by comparing its performance to a 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 [33]. The main difference is that we use Truncated Hierarchical Basis splines (THB-splines) [16], 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, 44, 7], and with IGA in [33].

The methods are implemented for multipatch geometries, using the IGA library G+Smo[1] and the optimization library Ipopt[2]. A benchmark study on the performance of different optimization algorithms in the context of structural optimization can be found in [38]. For a review of implementation aspects of IGA in general we refer the reader to [34] and for details of the G+Smo library we refer the reader to [26, 30]. The code used for this work can be found at https://github.com/gismo/shapeopt.

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 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:

$$\max_{\Omega\in\mathscr{O}_{ad}} E(\Omega,u), \tag{1a}$$

$$\text{s.t. } a_\Omega(u,v) = \ell_\Omega(v) \qquad \text{for all } v\in V, \tag{1b}$$

where $\mathscr{O}_{ad}$ 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 Ω and u will be approximated numerically using splines. Namely, we have $\Omega = G(]0,1[^d)$, with the *parametrization*

$$G(\boldsymbol{\xi}) = \sum_{i=1}^{N^g} \boldsymbol{c}_i R_i^g(\boldsymbol{\xi}), \tag{2}$$

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 use 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

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}^{\mathscr{b}} \\ \boldsymbol{c}^{\imath} \end{bmatrix}, \tag{3}$$

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

[1] https://github.com/gismo

[2] https://github.com/coin-or/Ipopt

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

$$u_h = \sum_{i=1}^{N} u_i R_i \circ G^{-1}, \tag{4}$$

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

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

As standard in the Galerkin approach, the elements of the stiffness matrix $K_{\boldsymbol{c}}$ and the load vector $f_{\boldsymbol{c}}$ are computed as $K_{\boldsymbol{c},i,j} = a_{G(]0,1[^d)}(R_j \circ G^{-1}, R_i \circ G^{-1})$, and $f_{\boldsymbol{c},i} = \ell_{G(]0,1[^d)}(R_i \circ G^{-1})$. 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 the 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 Ω into the parameter domain $]0,1[^d$. In particular, for all $\boldsymbol{\xi} \in]0,1[^d$ it is necessary that $\det(J(\boldsymbol{\xi})) > 0$. [3] A sufficient condition, which guarantees the validity of the parametrization, is discussed in Appendix A.

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 [33, 17]. Within this framework we formulate the optimization problem in terms of boundary control points $\boldsymbol{c}^{\mathscr{b}}$. The collection of interior control points $\boldsymbol{c}^{\mathscr{i}}$ for the geometry parametrization is treated as an implicit function of $\boldsymbol{c}^{\mathscr{b}}$, see Appendix B. Additionally, the parametrization validity constraints $\det(J) > 0$, 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

$$\min_{\Delta \boldsymbol{c}^{\mathscr{i}}} \frac{1}{2} \Delta \boldsymbol{c}^T H(\boldsymbol{c}_0) \Delta \boldsymbol{c} + \nabla W(\boldsymbol{c}_0)^T \Delta \boldsymbol{c} + W(\boldsymbol{c}_0), \tag{5}$$

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

$$H(\boldsymbol{c}_0) \Delta \boldsymbol{c} = -\nabla W(\boldsymbol{c}_0).$$

Using (3), this can be restated as

$$H_{c^{\mathscr{i}},c^{\mathscr{i}}} \Delta \boldsymbol{c}^{\mathscr{i}} = -\nabla W_{\boldsymbol{c}^{\mathscr{i}}} - H_{c^{\mathscr{i}},c^{\mathscr{b}}} \Delta \boldsymbol{c}^{\mathscr{b}}, \tag{6}$$

where $[H_{c^{\mathscr{i}},c^{\mathscr{i}}}]_{ij} = \partial^2 W / \partial {c^{\mathscr{i}}}_i \partial {c^{\mathscr{i}}}_j$, $[H_{c^{\mathscr{i}},c^{\mathscr{b}}}]_{ij} = \partial^2 W / \partial {c^{\mathscr{i}}}_i \partial {c^{\mathscr{b}}}_j$ and $[\nabla W_{c^{\mathscr{i}}}]_i = \partial W / \partial {c^{\mathscr{i}}}_i$. The new parametrization is then defined by the control points given by $\boldsymbol{c} = \boldsymbol{c}_0 + \Delta \boldsymbol{c}$.

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

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

$$
\begin{aligned}
\max_{\Delta \boldsymbol{c}^{\mathscr{b}}} \; & E(\boldsymbol{c}, \boldsymbol{u}), && \text{(7a)}\\
\text{s.t.} \; & K_{\boldsymbol{c}}\, \boldsymbol{u} = f_{\boldsymbol{c}}, && \text{(7b)}\\
& H_{c^{\mathscr{i}}, c^{\mathscr{i}}} \Delta \boldsymbol{c}^{\mathscr{i}} = -\nabla W_{\boldsymbol{c}^{\mathscr{i}}} - H_{c^{\mathscr{i}}, c^{\mathscr{b}}} \Delta \boldsymbol{c}^{\mathscr{b}}, && \text{(7c)}\\
& \boldsymbol{c} = \boldsymbol{c}_0 + \Delta \boldsymbol{c}, && \text{(7d)}\\
& \boldsymbol{d} \geq \varepsilon, && \text{(7e)}\\
& {\boldsymbol{c}^{\mathscr{b}}}_L \leq \boldsymbol{c}^{\mathscr{b}} \leq {\boldsymbol{c}^{\mathscr{b}}}_U, && \text{(7f)}
\end{aligned}
$$

where $\boldsymbol{c}_0$ is the reference parametrization, $\Delta \boldsymbol{c} = (\Delta \boldsymbol{c}^{\mathscr{b}}, \Delta \boldsymbol{c}^{\mathscr{i}})$, and (7e) is the sufficient condition for the validity of the parametrization discussed in Appendix A. Each time we solve the subproblem (7) we update the reference parametrization. In the model problem considered in this work we saw no further progress after 5 to 10 reparametrizations.

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 $\mathscr{S}_{\text{det}}$ where we compute expansion coefficients $\boldsymbol{d}$ of $\det J$. 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. [16]) implies that the spline control polygon converges locally to function values. Note that other locally refinable splines with this property are available, for example Polynomial splines over Hierarchical T-meshes (PHT-splines) [10] or Locally Refinable (LR B-splines) [24]. For a comparison of these methodologies we refer the reader to [27].

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

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 $\boldsymbol{c}^{\mathscr{i}}$ enter this formulation as independent optimization variables, in the same way as $\boldsymbol{c}^{\mathscr{b}}$. 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 the 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

[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.

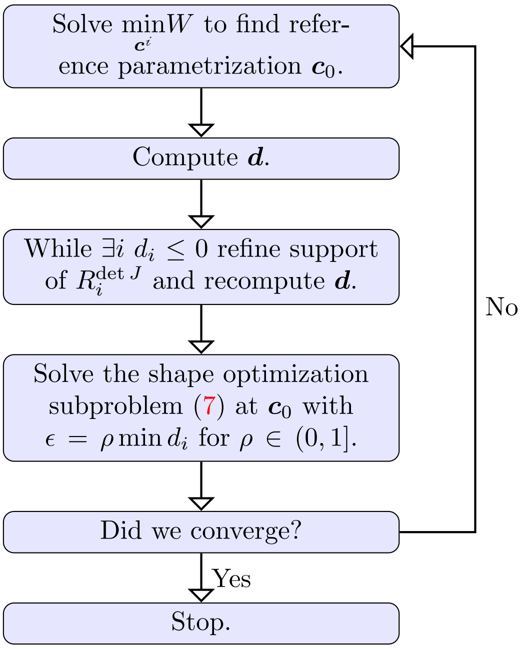

Solve $\min_{c'} W$ to find reference parametrization c_0.
Compute d.
While $\exists i\ d_i \le 0$ refine support of $R_i^{\det J}$ and recompute d.
Solve the shape optimization subproblem (7) at c_0 with $\epsilon = \rho \min d_i$ for $\rho \in (0,1]$.
Did we converge?
Stop.
No
Yes

Figure 1: Flowchart of the optimization algorithm.

idea has been used previously in the context of shape optimization in mechanics [13, 40]. Thus for a regularization parameter $\tau > 0$ we consider the optimization problem

$$\min_{\boldsymbol{c}} \tau W(\boldsymbol{c}) - E(\boldsymbol{c}, \boldsymbol{u}), \tag{8a}$$

$$\text{s.t. } K_{\boldsymbol{c}}\, \boldsymbol{u} = f_{\boldsymbol{c}}, \tag{8b}$$

$$\boldsymbol{c}_L \le \boldsymbol{c} \le \boldsymbol{c}_U. \tag{8c}$$

We put $W(\boldsymbol{c}) = \infty$ if $\det J \le 0$ at one of the quadrature points used for the integration when calculating $W(\boldsymbol{c})$. In this way, when the optimization algorithm enforces e.g. the standard sufficient decrease condition, such a choice ensures that the chosen step will always have positive determinant 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 τ 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 τ would lead to a compromise between these two extreme situations. One simple strategy for choosing such a value is to solve a sequence of problems (8) for decreasing values of τ.[5]

The two points above constitute the main drawback of this method. Namely that, in contrast to a constraint based method, we cannot guarantee that the parametrization is valid in between the quadrature points. Additionally one has to find a suitable value of the regularization parameter τ, which is problem-dependent.

[5] We should note that the literature on regularization is quite extensive, see for example [18] and references therein, and this topic is somewhat beyond the scope of this work.

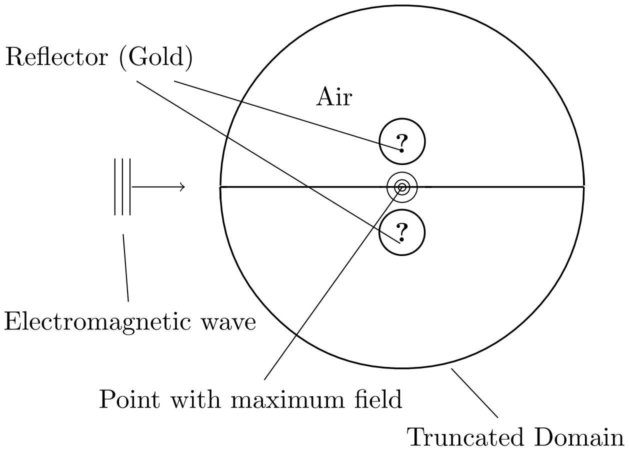


Figure 2: Sketch of the shape optimization problem. The goal is to find a shape of the reflector that maximizes the field close to a point.

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 ϵ_{cr} and μ_r denote the complex permittivity and permeability of the medium. Using the first order absorbing boundary condition [23] at the boundary Γ_t of the truncated domain, the electromagnetic field $\hat{u}$ should satisfy the following PDE:

$$\nabla \cdot \left(\frac{1}{\epsilon_{cr}} \nabla \hat{u} \right) + k_0^2 \mu_r \hat{u} = 0 \qquad \text{in } \Omega, \tag{9a}$$

$$\frac{\partial (\hat{u} - u^i)}{\partial n} + (jk_0 + \frac{1}{2r_t})(\hat{u} - u^i) = 0 \qquad \text{on } \Gamma_t. \tag{9b}$$

In the equations above, $k_0 = 2\pi\sqrt{\epsilon_0 \mu_0}$ is the wave number and ϵ_0, μ_0 refer to the permittivity and permeability, respectively, of free space. 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

$$u^i(x, y) = e^{-jk_0\sqrt{\epsilon_{cr}\mu_r}x}.$$

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 δ is a Gaussian bell-function

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

f	μ_r	μ_r^s	σ	ϵ_0	μ_0	$\epsilon_{r,\text{gold}}$
$4 \cdot 10^{14}[Hz]$	1.0	1.0	$10^6[S/m]$	$(\mu_0 c^2)^{-1}$	$4\pi 10^{-7}$	$-20.199 + j1.381$

Table 1: Physical parameters

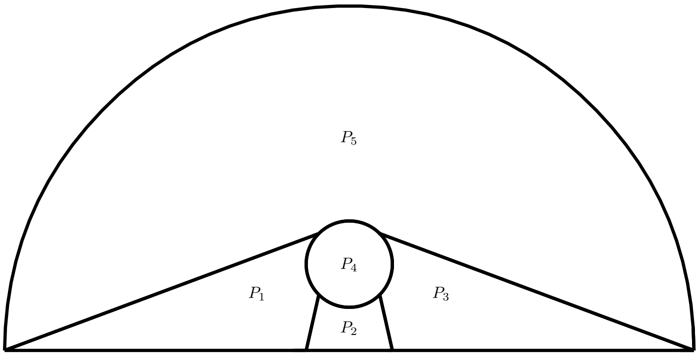


Figure 3: Patch layout.

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 $\epsilon_{cr}^s = \epsilon_{r,\text{gold}} - j\frac{\sigma}{\omega\epsilon_0}$.

The weak statement of the 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

$$\int_\Omega \frac{1}{\epsilon_{cr}} \hat{\nabla}\hat{u} \cdot \hat{\nabla}\hat{v} \mathrm{d}x + k_0^2 \int_\Omega \mu_r \hat{u}\hat{v} \mathrm{d}x + \left(jk_0 + \frac{1}{2r_t}\right) \int_{\Gamma_t} \frac{1}{\epsilon_{cr}} \hat{u}\hat{v} \mathrm{d}s \\ = \frac{1}{\epsilon_{cr}} \int_{\Gamma_t} \left(\frac{\partial u^i}{\partial n} + \left(jk_0 + \frac{1}{2r_t}\right) u^i \right) \hat{v} \mathrm{d}s. \quad (10)$$

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 dielectric 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 parametrized using splines of degree $p = 2$ as described in Section 2. We will use strong patch coupling to enforce C^0 continuity at the patch interfaces, while noting that other alternatives exists, for example weak coupling [21] and strong C^1 coupling [5].

The shape of the reflector is represented via the patch interfaces, which consist of 4 spline curves with C^0 continuity at the 4 corners. Such a description, for example, allows shapes that are only piecewise smooth such as e.g. the classical bowtie antenna [8]. There are other applications of IGA where even more complex geometries are relevant, for example when modelling crack propagation as in [37, 36].

Using the parametrizations we can pull back the equation (10) to the parameter domain and apply the Galerkin method to it, which ultimately results in a system of linear algebraic equations; see Appendix C for details. To evaluate the integrals involved we use element-wise Gauss-Legendre quadrature, owing to their immediate availability in the G+Smo library, while noting that more efficient alternatives exists [4, 2].

[6] For automatic generation of patch layouts the interested reader is referred to [49].

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 where the knotvectors used for representing the geometry are refined uniformly either 3 or 4 times.[7] 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 $\texttt{tol} = 10^{-3}$ 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 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 [8].

In our implementation we use the interior point solver Ipopt to solve the subproblems (7). As we solve a sequence of subproblems a warm start is available, namely the minimizer from the previous subproblem; so an interior point algorithm might not be the best choice of optimization algorithm [25]. However with the following parameter tuning we found the method to work well with Ipopt. 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 $\det J$ 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.

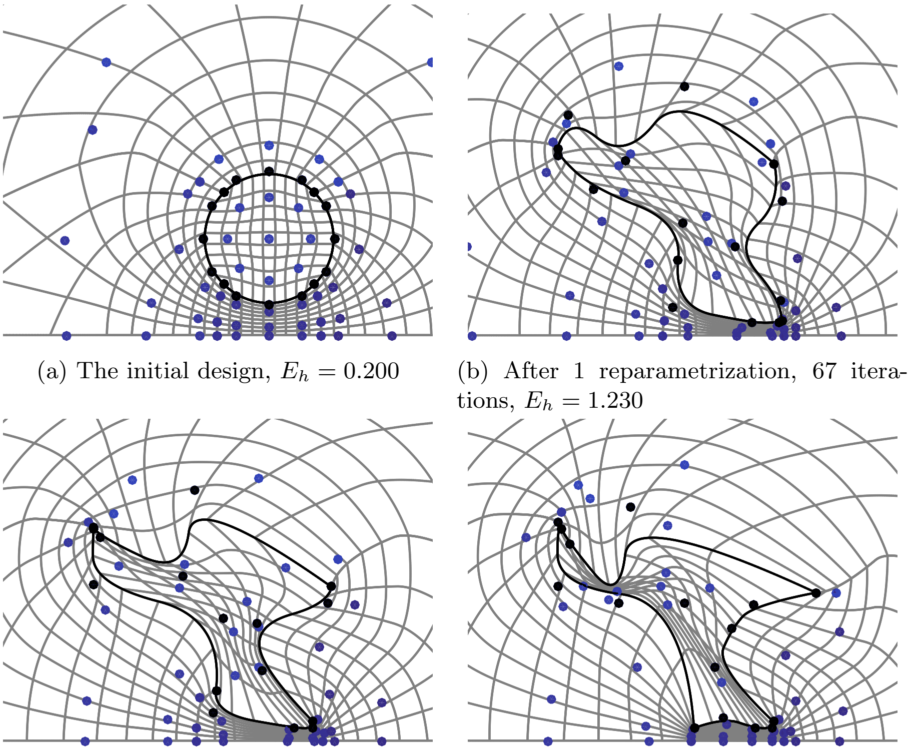
Another key parameter is the barrier parameter `mu_init`. Specifically, we use the `monotone` strategy, where the barrier parameter is monotonically decreased as the optimization algorithm progresses. However, if this parameter is too large in the beginning of the algorithm we found that it will push the design towards configurations with large $\det J$. To remedy this, instead of the default value 0.1 we use 10^{-4}. For more information about the optimization algorithm implemented in Ipopt and its parameters see [43].

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 $\det J$ 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

[7] Note that while we here use tensor product spline spaces for the approximation of the solution to the PDE, adaptive methods that use local refinement exist, see for example [16].

[8] 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 .



(a) The initial design, $E_h = 0.200$

(b) After 1 reparametrization, 67 iterations, $E_h = 1.230$

(c) After 2 reparametrizations, 143 iterations, $E_h = 1.484$

(d) After 10 reparametrizations, 579 iterations, $E_h = 1.803$

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.

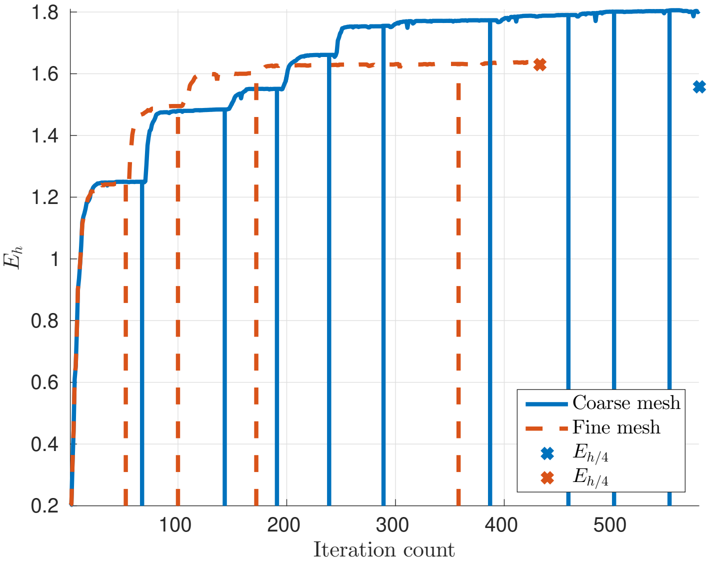
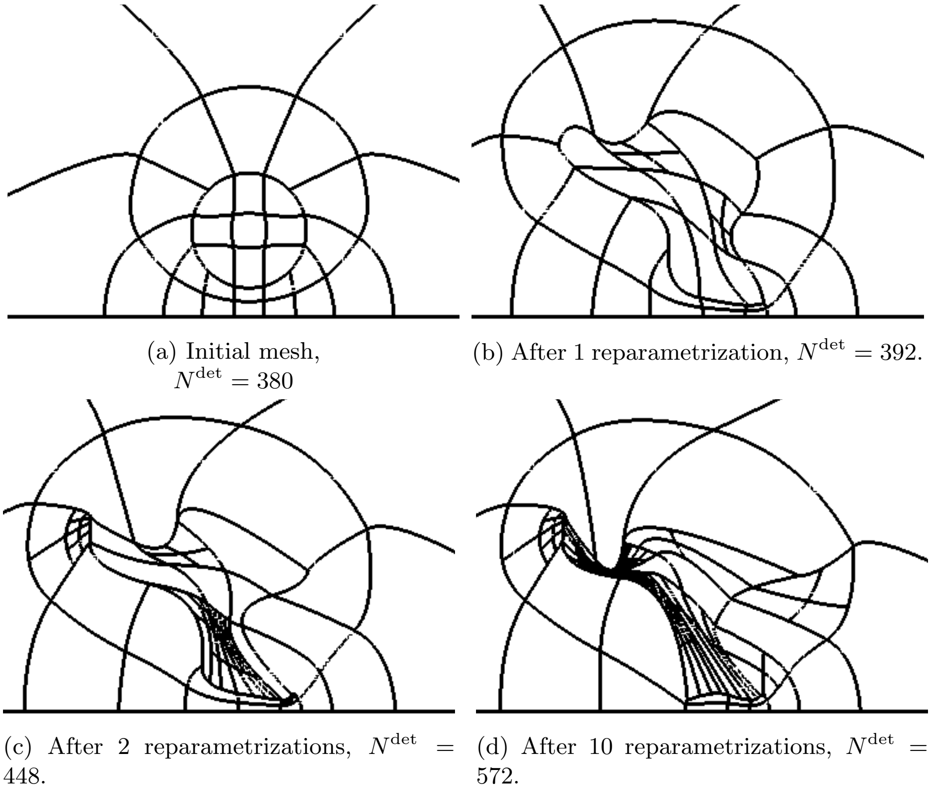


Figure 5: The objective function during optimization process for the coarse and fine meshes. The vertical lines indicate where the parametrization is updated. $E_{h/4}$ is the objective calculated on a refined mesh.



(a) Initial mesh, $N^{\mathrm{det}} = 380$

(b) After 1 reparametrization, $N^{\mathrm{det}} = 392$.

(c) After 2 reparametrizations, $N^{\mathrm{det}} = 448$.

(d) After 10 reparametrizations, $N^{\mathrm{det}} = 572$.

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

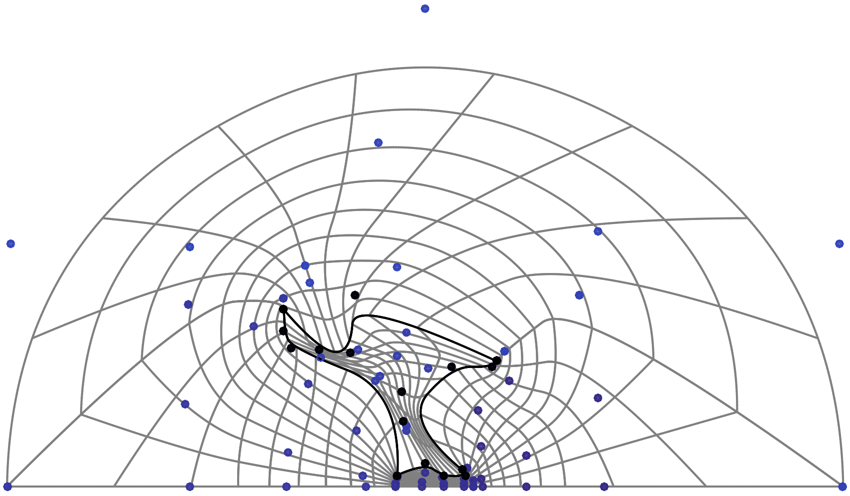


Figure 7: The final design using validity constraints and the fine mesh, $E_{h/2} = 1.638$

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 by 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

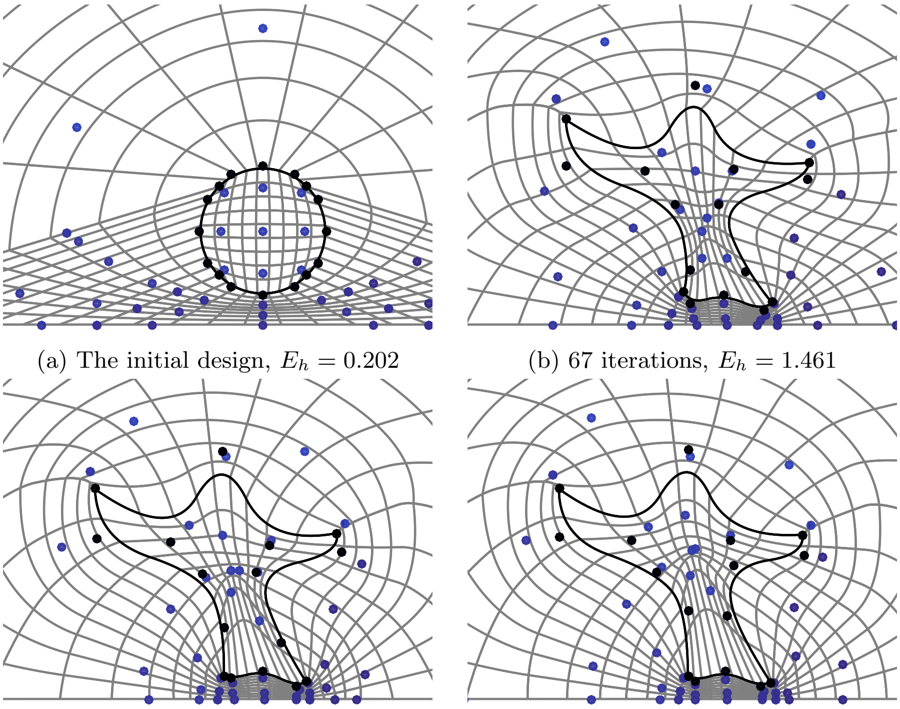
In this section we will present the results obtained with the method described in Section 4. With this method we perform the optimization with all control points as optimization variables while using the Winslow functional as a regularization term. Again we use the interior point solver Ipopt with a tolerance of `tol` $= 10^{-3}$ for 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 τW 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 the one 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 extent. 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



(a) The initial design, $E_h = 0.202$

(b) 67 iterations, $E_h = 1.461$

(c) 143 iterations, $E_h = 1.593$

(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.

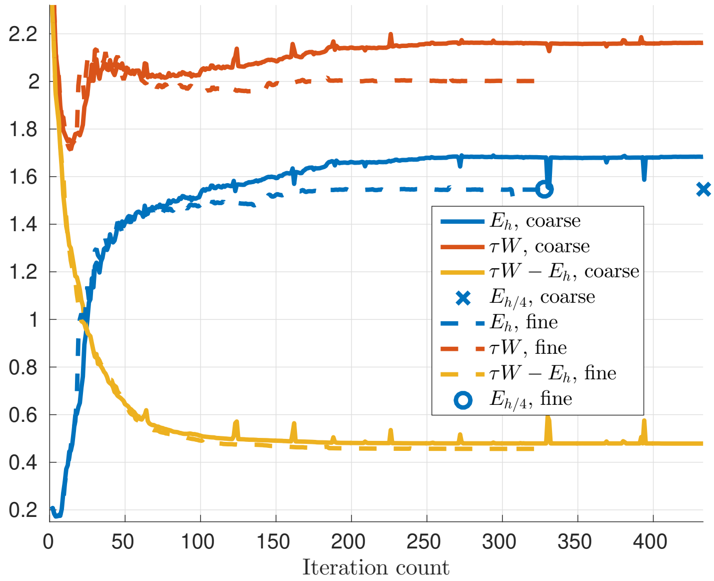


Figure 9: The objective function during the optimization process when using the regularization based approach, with the fine and coarse mesh.

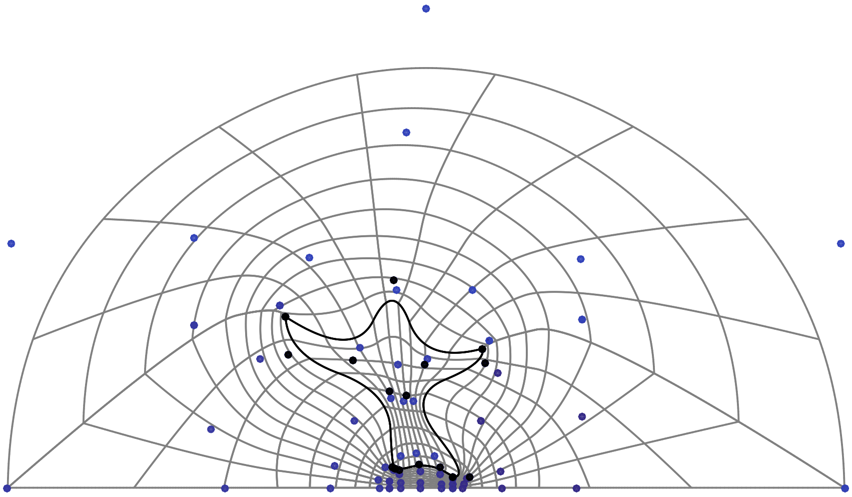


Figure 10: The final design when using the regularization approach and the fine mesh, $E_{h/2} = 1.545$.

result $E_{h/4} = 1.545$ with the difference at the 5th digit. The objective during the optimization process is plotted in Figure 9. The behaviour is similar for the two meshes, however the tolerance $\texttt{tol} = 10^{-3}$ 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. For the boundary-driven approach the computation of reference parametrizations by minimizing of the Winslow functional only accounts for 2.4% and 0.25% of the total running time for the coarse and the fine grid respectively.

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.

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.

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 uses a regularization term, and thus avoids both the validity constraints and the need for 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 2D 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. For obtaining competent overall running times, one can couple the approach with low-rank approximation techniques, which have been recently introduced in IGA [31, 41] and have proved efficient in the frame of PDE-constrained optimization [3].

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):228–248, 2010.

[2] M. Bartoň and V. M. Calo. Gauss–galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. *Computer-Aided Design*, 82:57–67, 2017.

[3] A. Bünger, S. Dolgov, and M. Stoll. A low-rank tensor method for pde-constrained optimization with isogeometric analysis. *SIAM Journal on Scientific Computing*, 42(1):A140–A161, 2020.

[4] F. Calabro, G. Sangalli, and M. Tani. Fast formation of isogeometric galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, 316:606–622, 2017.

[5] C. L. Chan, C. Anitescu, and T. Rabczuk. Strong multipatch c1-coupling for isogeometric analysis on 2d and 3d domains. *Computer Methods in Applied Mechanics and Engineering*, 357:112599, 2019.

[6] L. Chen, H. Lian, Z. Liu, H. Chen, E. Atroshchenko, and S. Bordas. Structural shape optimization of three dimensional acoustic problems with isogeometric boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 355:926–951, 2019.

[7] 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.

[8] R. Compton, R. McPhedran, Z. Popovic, G. Rebeiz, P. Tong, and D. Rutledge. Bow-tie antennas on a dielectric half-space: Theory and experiment. *IEEE transactions on antennas and propagation*, 35(6):622–631, 1987.

[9] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[10] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical t-meshes. *Graphical models*, 70(4):76–86, 2008.

[11] A. Falini, J. Špeh, and B. Jüttler. Planar domain parameterization with thb-splines. *Computer Aided Geometric Design*, 35-36:95 – 108, 2015. Geometric Modeling and Processing 2015.

[12] G. Farin and D. Hansford. Discrete coons patches. *Computer aided geometric design*, 16(7):691–700, 1999.

[13] J. Friederich, M. Scherer, and P. Steinmann. Isogeometric structural shape optimization using a fictitious energy regularization. *PAMM*, 11(1):709–710, 2011.

[14] P. Gangl, U. Langer, A. Mantzaflaris, and R. Schneckenleitner. Isogeometric simulation and shape optimization with applications to electrical machines. In G. Nicosia and V. Romano, editors, *Scientific Computing in Electrical Engineering*, pages 35–43, Cham, 2020. Springer International Publishing.

[15] H. Ghasemi, H. S. Park, and T. Rabczuk. A level-set based iga formulation for topology optimization of flexoelectric materials. *Computer Methods in Applied Mechanics and Engineering*, 313:239–258, 2017.

[16] C. Giannelli, B. Jüttler, and H. Speleers. Thb-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[17] 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.

[18] P. C. Hansen. Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems. *Numerical algorithms*, 6(1):1–35, 1994.

[19] 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.

[20] 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.

[21] Q. Hu, F. Chouly, P. Hu, G. Cheng, and S. P. Bordas. Skew-symmetric nitsche's formulation in isogeometric analysis: Dirichlet and symmetry conditions, patch coupling and frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 341:188–220, 2018.

[22] 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):4135–4195, 2005.

[23] J.-M. Jin. *The finite element method in electromagnetics.* Wiley, 1993.

[24] K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using lr b-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.

[25] E. John and E. A. Yıldırım. Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41(2):151–183, 2008.

[26] B. Jüttler, U. Langer, A. Mantzaflaris, S. E. Moore, and W. Zulehner. Geometry+ simulation modules: Implementing isogeometric analysis. *PAMM*, 14(1):961–962, 2014.

[27] X. Li, F. Chen, H. Kang, and J. Deng. A survey on the local refinable splines. *Science China Mathematics*, 59(4):617–644, 2016.

[28] C. Liu, L. Chen, W. Zhao, and H. Chen. Shape optimization of sound barrier using an isogeometric fast multipole boundary element method in two dimensions. *Engineering Analysis with Boundary Elements*, 85:142–157, 2017.

[29] J. López, C. Anitescu, and T. Rabczuk. Cad-compatible structural shape optimization with a movable bézier tetrahedral mesh. *Computer Methods in Applied Mechanics and Engineering*, 367:113066, 2020.

[30] A. Mantzaflaris. An overview of geometry plus simulation modules. In D. Slamanig, E. Tsigaridas, and Z. Zafeirakopoulos, editors, *Mathematical Aspects of Computer and Information Sciences*, pages 453–456, Cham, 2020. Springer International Publishing.

[31] 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.

[32] 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.

[33] D. Nguyen, A. Evgrafov, and J. Gravesen. Isogeometric shape optimization for electromagnetic scattering problems. *Progress in Electromagnetics Research B*, 45:117–146, 2012.

[34] V. P. Nguyen, C. Anitescu, S. P. Bordas, and T. Rabczuk. Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117:89–116, 2015.

[35] M. Pan and F. Chen. Low-rank parameterization of volumetric domains for isogeometric analysis. *Computer-Aided Design*, 114:82–90, 2019.

[36] X. Peng, E. Atroshchenko, P. Kerfriden, and S. P. A. Bordas. Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth. *Computer Methods in Applied Mechanics and Engineering*, 316:151–185, 2017.

[37] X. Peng, E. Atroshchenko, P. Kerfriden, and S. P. A. Bordas. Linear elastic fracture simulation directly from cad: 2d nurbs-based implementation and role of tip enrichment. *International Journal of Fracture*, 204(1):55–78, 2017.

[38] S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization*, 52(3):527–547, 2015.

[39] 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.

[40] 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.

[41] F. Scholz, A. Mantzaflaris, and B. Juettler. Partial tensor decomposition for decoupling isogeometric Galerkin discretizations. *Computer Methods in Applied Mechanics and Engineering*, 336:485 – 506, 2018.

[42] A. Shamanskiy, M. H. Gfrerer, J. Hinz, and B. Simeon. Isogeometric parametrization inspired by large elastic deformation. *arXiv preprint arXiv:1810.12425*, 2018.

[43] 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.

[44] E. Wadbro and C. Engström. Topology and shape optimization of plasmonic nano-antennas. *Computer Methods in Applied Mechanics and Engineering*, 293:155–169, 2015.

[45] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of computational physics*, 1(2):149–172, 1966.

[46] 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.

[47] 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.

[48] 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.

[49] 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 $\det(J) > 0$ 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 $\det J \in C^{k-1}$ is a spline with degree $d \cdot p - 1$. This means that we can construct a spline space $\mathcal{S}_{\text{det}}$ that contains $\det J$.[9] We can therefore find the expansion coefficients $\boldsymbol{d}$ of $\det J$ with respect to B-splines in this space.[10]

[9] Specifically, the smallest spline space containing $\det J$ 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.

[10] 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. [31]) therefore the solution can be obtained extremely fast.

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

$$\boldsymbol{d} \geq \varepsilon, \tag{11}$$

where ε is a small positive algorithmic parameter. This condition guarantees that $\det J > 0$ 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 $\det J(\boldsymbol{\xi}) > 0$, $\forall \boldsymbol{\xi} \in]0,1[^d$. 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 ε is small enough and $\det J > 0$ 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 for 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

B.1 Coons patch and spring method

Two simple methods for constructing a grid of control points are the Coons patch [12] and the spring method [17]. 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 optimization–based methods. Here the geometry map G is chosen such that it minimizes a quality metric $w(\boldsymbol{\xi})$

$$\min_{G} \int_{]0,1[^d} w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{12}$$

$$\text{s.t. } G|_{\partial[0,1]^d} = \gamma, \tag{13}$$

where γ is a given boundary curve.

There are several different quality metrics to choose from, cf. [17, 48]. In this work we will consider the Winslow functional which is given in terms of the Jacobian matrix J as

$$W = \int_{]0,1[^d} w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{14}$$

with

$$w = \frac{\operatorname{tr}\left(J^T J\right)}{\det J}. \tag{15}$$

In 2D the Winslow functional has the nice property that its minimizer has a harmonic inverse [17]. 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 α is given by

$$\frac{\partial w}{\partial \alpha} = 2(\det J)^{-1}\mathrm{tr}\left(J^T \frac{\partial J}{\partial \alpha}\right) - \mathrm{tr}\left(J^{-1}\frac{\partial J}{\partial \alpha}\right)\frac{\mathrm{tr}(J^T J)}{\det J}, \tag{16}$$

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

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

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 α, which is the case when α 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:

$$\begin{aligned} &\int_{]0,1[^2} \frac{1}{\epsilon_{cr}} J^{-T}\nabla u \cdot J^{-T}\nabla v \, |\det J| \, \mathrm{d}\boldsymbol{\xi} - k_0^2 \int_{]0,1[^2} \mu_r u v \, |\det J| \, \mathrm{d}\boldsymbol{\xi} \\ &\qquad + \left(jk_0 + \frac{1}{2r_t}\right)\int_{G^{-1}(\Gamma_t)} \frac{1}{\epsilon_{cr}} u v \left|\frac{\partial G}{\partial t}\right| \mathrm{d}t \\ &\qquad = \frac{1}{\epsilon_{cr}} \int_{G^{-1}(\Gamma_t)} \frac{1}{\epsilon_{cr}} \left(\frac{\partial u^i}{\partial n}\circ G + \left(jk_0 + \frac{1}{2r_t}\right) u^i \circ G\right) v \left|\frac{\partial G}{\partial t}\right| \mathrm{d}t \end{aligned} \tag{18}$$

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

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

where $\boldsymbol{u} = (u_1, \ldots, u_N)^T$ and K, M, T and f is given by

$$K_{kl} = \int_{[0,1]^2} \frac{1}{\epsilon_{cr}} J^{-T} \nabla R_k \cdot J^{-T} \nabla R_l \left|\det J\right| \, \mathrm{d}\boldsymbol{\xi}, \tag{20a}$$

$$M_{kl} = -k_0^2 \int_{[0,1]^2} \mu_r R_k R_l \left|\det J\right| \, \mathrm{d}\boldsymbol{\xi}, \tag{20b}$$

$$T_{kl} = \left(jk_0 + \frac{1}{2r_t} \right) \int_{G^{-1}(\Gamma_t)} \frac{1}{\epsilon_{cr}} R_k R_l \left| \frac{\partial G}{\partial t} \right| \, \mathrm{d}t, \tag{20c}$$

$$f_l = \int_{G^{-1}(\Gamma_t)} \frac{1}{\epsilon_{cr}} \left(\frac{\partial u^i}{\partial n} \circ G + \left(jk_0 + \frac{1}{2r_t} \right) u^i \circ G \right) e_l \left| \frac{\partial G}{\partial t} \right| \, \mathrm{d}\xi. \tag{20d}$$

Note that the values of ε_{cr} and μ_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

$$\begin{bmatrix} \Re(A) & -\Im(A) \\ -\Im(A) & -\Re(A) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{u}) \\ \Im(\boldsymbol{u}) \end{bmatrix} = \begin{bmatrix} \Re(\boldsymbol{f}) \\ -\Im(\boldsymbol{f}) \end{bmatrix}.$$