DE GRUYTER Comput. Methods Appl. Math. 2015; 15 (3):291–305

Research Article

Ralf Hiptmair and Alberto Paganini

Shape Optimization by Pursuing Diffeomorphisms

Abstract: We consider PDE constrained shape optimization in the framework of finite element discretization of the underlying boundary value problem. We present an algorithm tailored to preserve and exploit the approximation properties of the finite element method, and that allows for arbitrarily high resolution of shapes. It employs (i) B-spline based representations of the deformation diffeomorphism, and (ii) superconvergent domain integral expressions for the shape gradient. We provide numerical evidence of the performance of this method both on prototypical well-posed and ill-posed shape optimization problems.

Keywords: Shape Optimization, PDE Constraint, Finite Element Method

MSC 2010: 40Q10, 49K20, 65N30

DOI: 10.1515/cmam-2015-0013
Received February 6, 2015; revised April 29, 2015; accepted May 7, 2015

1 Introduction

Physical phenomena are described by mathematical models that link input and output quantities. An important task in engineering is to find optimal values of the input so that a target output is minimized. In shape optimization the target output depends on the shape Ω of an object. This dependence is modeled via a shape functional $\mathcal{J}$.

In several relevant applications the shape functional $\mathcal{J}$ depends, additionally, on the solution of a boundary value problem (BVP) stated on Ω. In this case we speak of PDE constrained shape optimization. These optimization problems are highly non-linear and can rarely be solved analytically. Usually, one has to content oneself with approximate optimal shapes obtained with iterative optimization algorithms combined with approximate solutions of the underlying BVP. Clearly, the quality of the approximate optimal shapes heavily depends on the choice of the numerical method used to retrieve them.

An accurate method to solve PDE constrained optimization problems has been developed relying on boundary element method solutions of the underlying BVP [16, 17]. However, the bulk of literature considers discretizations by means of the finite element method (FEM) [2, 6–8, 18, 25–28, 30]. In this case we can distinguish between moving-mesh and fixed-mesh methods.

The former discretize an initial guess Ω_0 with a mesh and then optimize the coordinates of the mesh nodes [2, 28, 30]. This is a very delicate task because the mesh might get distorted or self-intersect as the optimization routine proceeds [3, 4].

Among the fixed-mesh methods, the two most popular approaches are level-set methods and free-form deformation methods. In the level-set approach, the boundary of the optimal domain is represented as the zero-level of a function [5]. The optimization is then carried out by updating this function. Again, this is a delicate process because, to identify the boundary of the optimized domain, the level set function should have steep slope at the zero-level. However, as the optimization proceeds, it is observed that level functions tend to become flat [29].

Ralf Hiptmair, Alberto Paganini: Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland, e-mail: hiptmair@sam.math.ethz.ch, alberto.paganini@sam.math.ethz.ch

On the other hand, free-form deformation methods [7, 26] recast the shape optimization problem as an optimal control problem. Shapes are parametrized by applying a transformation to the initial guess Ω_0. This transformation is constructed with (piecewise) polynomials defined on a lattice of control points, and optimization is carried out on their coordinates. This approach allows to preserve the approximation properties of FEM. However, the infinite dimensional shape optimization problem is replaced with a counterpart with a fixed small number of control parameters, and the dependence of the quality of the discrete solution on the number of control parameters is not clear.

We present an algorithm developed to preserve and exploit the approximation properties of FEM, and that allows for arbitrarily high resolution of shapes. Similar to the free-form deformation approach, we recast the shape optimization problem as an optimal control problem. Shapes are parametrized by letting a diffeomorphism act on an initial shape Ω_0. Pursuing a Ritz approach, we discretize the diffeomorphism with conforming basis functions based on cubic B-splines. We show that, under reasonable assumptions, the sequence of optimal discrete solutions converges to the global minimum as the dimension of the trial space tends to infinity. We also investigate the impact of FEM approximations in the context of elliptic PDE constrained shape optimization and formulate a descent method that enjoys superconvergence in the approximation of the Fréchet derivative. We test the performance of the proposed method both on a well-posed model problem stemming from the class of exterior Bernoulli free boundary problems and on a prototypical ill-posed inverse problem.

2 Shape Optimization in Parametric Form

Let $D \subset \mathbb{R}^d$ be bounded and convex domain (hold-all domain), and let Ω_0 be a compact subset of D with Lipschitz boundary. We fix $\varepsilon > 0$ and define the set of admissible shapes as

$$\mathcal{U}_{\mathrm{ad}}(\Omega_0) := \{T_{\mathcal{V}}(\Omega_0);\ T_{\mathcal{V}} := \mathcal{I} + \mathcal{V}, \|\mathcal{V}\|_{C^1(\overline{D};\mathbb{R}^d)} \le 1 - \varepsilon\}. \tag{2.1}$$

Note that the map $T_{\mathcal{V}} := \mathcal{I} + \mathcal{V}$ is a diffeomorphism whenever $\|\mathcal{V}\|_{C^1(\overline{D};\mathbb{R}^d)} < 1$ (see [2, Lemma 6.13]). Let $\mathcal{J}$ be a real-valued functional defined on $\mathcal{U}_{\mathrm{ad}}(\Omega_0)$, and let $\tilde{\mathcal{J}}$ be defined by

$$\tilde{\mathcal{J}} : B^1_{1-\varepsilon} \to \mathbb{R}, \quad \mathcal{V} \mapsto \mathcal{J}(T_{\mathcal{V}}(\Omega_0)),$$

where $B^k_{1-\varepsilon}$ denotes the closed ball in $C^k(\overline{D};\mathbb{R}^d)$ of radius $1-\varepsilon$ centered in 0. The shape optimization problem

$$\inf_{\Omega \in \mathcal{U}_{\mathrm{ad}}(\Omega_0)} \mathcal{J}(\Omega)$$

can be recast as

$$\inf_{\mathcal{V} \in B^1_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}). \tag{2.2}$$

Theorem 2.1. *Let $\tilde{\mathcal{J}}$ be continuous with respect to the $C^1(\overline{D};\mathbb{R}^d)$- norm and restrict the shape optimization problem* (2.2) *to*

$$\inf_{\mathcal{V} \in B^2_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}). \tag{2.3}$$

Then, there exists a vector field $\mathcal{V}^* \in C^1(\overline{D};\mathbb{R}^d)$ so that

$$\tilde{\mathcal{J}}(\mathcal{V}^*) = \inf_{\mathcal{V} \in B^2_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}).$$

Proof. We follow closely [2, Theorem 5.12]. The main ingredient is the compact embedding

$$C^2(\overline{D};\mathbb{R}^d) \overset{c}{\hookrightarrow} C^1(\overline{D};\mathbb{R}^d),$$

which holds for D convex or, more generally, if "every pair of points $x, y \in D$ can be joined with a rectifiable arc in D having length not exceeding some fixed multiple of $|x-y|$" [1, Theorem 1.34].

A minimizing sequence of (2.3) is bounded (by definition of the optimization problem). Thus, by compactness, we can extract a subsequence that converges to a limit function $\hat{\mathcal{V}}$ in the $C^1(\overline{D};\mathbb{R}^d)$-norm. Finally, the continuity assumption on $\tilde{\mathcal{J}}$ implies $\hat{\mathcal{V}} = \mathcal{V}^*$. □

Remark 2.2. The continuity assumption on $\tilde{\mathcal{J}}$ in Theorem 2.1 is fulfilled by most of the shape functionals considered in literature. For instance, this is the case for the volume and the surface area shape functionals.

Remark 2.3. A counterpart of Theorem 2.1 still holds if the function spaces $C^1(\overline{D};\mathbb{R}^d)$, $C^2(\overline{D};\mathbb{R}^d)$ are replaced by $W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$, $W^{2,\infty}(\mathbb{R}^d;\mathbb{R}^d)$, respectively. However, having approximations by means of the Ritz method in mind, we restrict our framework to separable spaces.

Remark 2.4. There is little hope for uniqueness in this framework. Let $\mathcal{V}^*$ be an optimal solution. If there is a vector field $\tilde{\mathcal{V}} \neq 0$ so that $(\mathcal{I} + \tilde{\mathcal{V}})(\partial\Omega_0) = \partial\Omega_0$ (from the set point of view), then the composition $\mathcal{V}^* \circ (\mathcal{I} + \tilde{\mathcal{V}})$ is an optimal solution, too.

Approximate solutions can be obtained easily with a Ritz approach.

Theorem 2.5. *Let $\{V_N\}_{N\in\mathbb{N}}$ be a nested sequence of $C^2(\overline{D};\mathbb{R}^d)$-conforming trial spaces that satisfies*

$$\overline{\bigcup_{N\in\mathbb{N}} V_N}^{C^2(\overline{D};\mathbb{R}^d)} = C^2(\overline{D};\mathbb{R}^d).$$

Let $\{\mathcal{V}_N^*\}_{N\in\mathbb{N}}$ be the sequence of discrete solutions defined by

$$\mathcal{V}_N^* \in \operatorname*{argmin}_{\mathcal{V}_N\in V_N\cap B^2_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}_N). \tag{2.4}$$

Then, under the assumptions of Theorem 2.1, $\{\mathcal{V}_N^*\}_{N\in\mathbb{N}}$ is a minimizing sequence of $\tilde{\mathcal{J}}$.

Proof. We follow closely the proof of the classic result on the convergence of Ritz methods given in [20, Section 40.1]. Let $\mu \in \mathbb{R}$ be the infimum of (2.3). Note that $\mu > -\infty$. Let $a > 0$, and let $\mathcal{V} \in B^2_{1-\varepsilon}$ satisfy

$$\tilde{\mathcal{J}}(\mathcal{V}) < \mu + a.$$

By continuity of $\tilde{\mathcal{J}}$, the vector field $\mathcal{V}$ can be rescaled so that

$$\|\mathcal{V}\|_{C^2(\overline{D};\mathbb{R}^d)} < 1 - \varepsilon \quad \text{and} \quad \tilde{\mathcal{J}}(\mathcal{V}) < \mu + 2a.$$

Let $b > 0$, and let $N = N(b) \in \mathbb{N}$ be sufficiently large. Then, there exists a $\mathcal{V}_N \in V_N \cap B^2_{1-\varepsilon}$ that satisfies

$$\|\mathcal{V} - \mathcal{V}_N\|_{C^2(\overline{D};\mathbb{R}^d)} < b.$$

Furthermore, for $b = b(a)$ small enough, it holds

$$\tilde{\mathcal{J}}(\mathcal{V}_N) < \mu + 3a.$$

Let $\mathcal{V}_N^*$ be defined as in (2.4). It holds

$$\mu \le \tilde{\mathcal{J}}(\mathcal{V}_N^*) \le \tilde{\mathcal{J}}(\mathcal{V}_N) \le \mu + 3a.$$

Since a is arbitrary, it follows

$$\lim_{N\to\infty} \tilde{\mathcal{J}}(\mathcal{V}_N^*) = \mu.$$ □

Remark 2.6. Refined convergence theories can be found in [17, 18, 25]. These articles rely on a parametrization of the boundary, and consider as admissible shapes those that can be reached via a normal perturbation of the boundary $\partial\Omega_0$. In this case, the parametrization of shapes is unique, and a priori convergence rates can be proved.

3 PDE Constrained Shape Optimization

In PDE constrained shape optimization, the goal is to find the domain Ω that minimizes the functional $\mathcal{J}(\Omega, u)$ subject to a PDE constraint $\mathcal{A}u = f$ in Ω. Here, $\mathcal{A} : X(\Omega) \to X(\Omega)^*$ denotes a second order $X(\Omega)$-elliptic operator between the Hilbert space $X(\Omega)$ and its dual $X(\Omega)^*$, which are function spaces on the domain Ω. Similarly as in (2.2), the shape optimization problem can be recast in a parametric form relying on the characterization of admissible domains (2.1), that is,

$$\inf_{\mathcal{V}\in B^1_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}, u) \quad \text{subject to} \quad \tilde{\mathcal{A}}_{\mathcal{V}} u = \tilde{f}_{\mathcal{V}} \text{ in } \Omega_0. \tag{3.1}$$

Both the elliptic operator $\tilde{\mathcal{A}}_{\mathcal{V}} : X(\Omega_0) \to X(\Omega_0)^*$ and the linear functional $\tilde{f}_{\mathcal{V}} \in X(\Omega_0)^*$ depend on the vector field $\mathcal{V}$ and are created in a way so that $u \in X(\Omega_0)$ is the solution to $\tilde{\mathcal{A}}_{\mathcal{V}} u = \tilde{f}_{\mathcal{V}}$ in Ω_0 if and only if $\hat{u} := u \circ T_{\mathcal{V}}^{-1} \in X(\Omega)$ is the solution to $\mathcal{A}\hat{u} = f$ in Ω.

The idea of transforming both the shape functional and the PDE constraint on a reference domain is not new to shape optimization. It has already been used, for instance, in [8, 18, 27, 28], and is, de facto, the standard approach for shape optimization based on free-form deformations; see [7, 26] and references therein.

Example 3.1. The parametric form of the shape optimization problem

$$\inf_{\Omega\in\mathcal{U}_{\mathrm{ad}}(\Omega_0)} \mathcal{J}(\Omega, \hat{u}) \quad \text{subject to} \quad \begin{cases} -\Delta\hat{u} = f & \text{in } \Omega, \\ \hat{u} = 0 & \text{on } \partial\Omega, \end{cases} \tag{3.2}$$

with $\mathcal{J}(\Omega, \hat{u}) := \int_\Omega j(\hat{u})\,\mathrm{d}\boldsymbol{x}$, $j \in C^1(\mathbb{R})$, and $f \in H^{-1}(\Omega)$, reads

$$\inf_{\mathcal{V}\in B^1_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}, u) \quad \text{subject to} \quad \begin{cases} -\operatorname{div}(\mathbf{M}_{\mathcal{V}} \operatorname{grad} u) = (\det \mathbf{D}T_{\mathcal{V}}) T_{\mathcal{V}}^*(f) & \text{in } \Omega_0, \\ u = 0 & \text{on } \partial\Omega_0, \end{cases} \tag{3.3}$$

where the pullback $T_{\mathcal{V}}^*$ is defined as the composition $T_{\mathcal{V}}^*(f) := f \circ T_{\mathcal{V}}$,

$$\tilde{\mathcal{J}}(\mathcal{V}, u) := \int_{\Omega_0} j(u)(\det \mathbf{D}T_{\mathcal{V}})\,\mathrm{d}\boldsymbol{x} \quad \text{and} \quad \mathbf{M}_{\mathcal{V}} := (\det \mathbf{D}T_{\mathcal{V}})\mathbf{D}T_{\mathcal{V}}^{-1}\mathbf{D}T_{\mathcal{V}}^{-T}.$$

Assuming continuity of the map $\mathcal{V} \mapsto \tilde{\mathcal{J}}(\mathcal{V}, u)$ on $C^1(\overline{D}; \mathbb{R}^d)$, an approximate solution of (3.1) can be obtained as in Theorem 2.5 by computing

$$\mathcal{V}_N^* \in \operatorname*{argmin}_{\mathcal{V}_N\in V_N\cap B^2_{1-\varepsilon}} \tilde{\mathcal{J}}(\mathcal{V}_N, u) \quad \text{subject to} \quad \tilde{\mathcal{A}}_{\mathcal{V}_N} u = \tilde{f}_{\mathcal{V}_N} \text{ in } \Omega_0 \tag{3.4}$$

for N large enough. Note that the approximate optimal solution $\mathcal{V}_N^*$ must satisfy the variational inequality [22, Theorem 1.48]

$$d\tilde{\mathcal{J}}(\mathcal{V}_N^*, u; \mathcal{W}_N - \mathcal{V}_N^*) \geq 0 \quad \text{for all } \mathcal{W}_N \in V_N \cap B^2_{1-\varepsilon}, \tag{3.5}$$

where $d\tilde{\mathcal{J}}$ denotes the Fréchet derivative of $\tilde{\mathcal{J}}$.

Remark 3.2. In Example 3.1, a minimizing sequence $\{\mathcal{V}_N^*\}_{N\in\mathbb{N}}$ that satisfies (3.4) contains a subsequence $\{\mathcal{V}_{N_i}^*\}_{i\in\mathbb{N}}$ that converges strongly in $C^1(\overline{D}; \mathbb{R}^d)$ to a $\hat{\mathcal{V}} \in B^1_{1-\varepsilon}$. Therefore, the ellipticity constants of $\{\tilde{\mathcal{A}}_{\mathcal{V}_{N_i}}\}_{i\in\mathbb{N}}$ are bounded from below by a constant $c > 0$. This implies that

$$\|u_{\hat{\mathcal{V}}} - u_{N_i}\|_{H^1(\Omega_0)} \to 0 \quad \text{as } i \to \infty,$$

where u_{N_i} is the solution to $\tilde{A}_{\mathcal{V}_{N_i}} u = \tilde{f}_{\mathcal{V}_{N_i}}$ and $u_{\hat{\mathcal{V}}}$ is the solution to $\tilde{A}_{\hat{\mathcal{V}}} u = \tilde{f}_{\hat{\mathcal{V}}}$, see [2, Lemma 5.3]. With this result it is easy to show $C^1(\overline{D}; \mathbb{R}^d)$-continuity of the constraint functional (3.3).

4 Algorithm

We focus on the optimization problem (3.4). Let $\Omega_0 \subset D$ be an initial guess. As trial space V_N, we choose the space spanned by multivariate B-splines of degree 3 on a regular grid that covers the hold-all domain D; see [24, Section 7.3]. Note that the hold-all domain D can be chosen to have a simple shape, e.g., a tensor product domain. More precisely, vector fields belonging to V_n can be written as

$$\mathcal{V}_N(\boldsymbol{x}) = \sum_{i=1}^{N} \Big(\sum_{j=1}^{d} c_i^j \mathbf{e}_j \Big) \mathbf{B}_i(\boldsymbol{x}), \quad c_i^j \in \mathbb{R}, \tag{4.1}$$

where $\mathbf{B}_i$ denotes the i-th multivariate B-spline of degree 3, and $\mathbf{e}_j$, $j = 1, \dots, d$, are basis vectors of $\mathbb{R}^d$.

The trial space V_N fulfills the assumptions of Theorem 2.5: it contains tensorized polynomials by Marsden's identity [24, Section 4.3], and it is $C^2(\overline{D}; \mathbb{R}^d)$-conforming because multivariate B-splines of degree 3 are twice continuously differentiable by construction. Moreover, B-splines have compact support and are polynomial in each grid cell. These two properties are crucial for an efficient implementation of the algorithm. Finally, using B-splines defined on a regular grid greatly simplifies the implementation, because every B-spline $\mathbf{B}_i$ is obtained by translating a single "mother" function [24, Section 7.3].

An approximation of the discrete optimal solution $\mathcal{V}_N^*$ can be retrieved with descent methods, which rely on the Fréchet derivative $d\tilde{\jmath}$ of $\tilde{\jmath}$ and are guaranteed to converge to (local) minima by the compactness of $V_N \cap B^2_{1-\varepsilon}$. Formulas for the Fréchet derivative of $\tilde{\jmath}$ can easily be derived with the Lagrangian approach described in [22, Section 1.6.4]. Note that this approach is simpler than the Lagrangian approach for deriving the Eulerian derivative of $\jmath(\Omega, u)$ described in [14]; indeed, in the parametric approach described in Section 3, the function space to which u belongs is independent of the control parameter $\mathcal{V}$.

Remark 4.1. The Fréchet derivative of $\tilde{\jmath}(\cdot, u)$ at $\mathcal{V}$ evaluated in the direction $\mathcal{W}$ is equal to the Eulerian derivative of $\jmath(T_{\mathcal{V}}(\Omega), u)$ in the direction $\mathcal{W} \circ (T_{\mathcal{V}}^{-1})$, because $T_{\mathcal{V}+\mathcal{W}} = T_{\mathcal{W} \circ T_{\mathcal{V}}^{-1}} \circ T_{\mathcal{V}}$.

Example 4.2. The Fréchet derivative of $\tilde{\jmath}$ from (3.3) reads

$$d\tilde{\jmath}(\mathcal{V}, u; \mathcal{W}) = \int_{\Omega_0} (j(u) - fp) \partial_{\mathcal{W}}(\det \mathbf{D}T_{\mathcal{V}}) - \operatorname{grad} f \cdot \mathcal{W} p \det \mathbf{D}T_{\mathcal{V}} + \operatorname{grad} p \cdot \partial_{\mathcal{W}} \mathbf{M}_{\mathcal{V}} \operatorname{grad} u \, \mathrm{d}\boldsymbol{x},$$

where

$$\begin{aligned} \partial_{\mathcal{W}} \mathbf{M}_{\mathcal{V}} &:= \det(\mathbf{D}T_{\mathcal{V}}) (\operatorname{tr}(\mathbf{D}T_{\mathcal{V}}^{-1} \mathbf{D}\mathcal{W}) \mathbf{D}T_{\mathcal{V}}^{-1} \mathbf{D}T_{\mathcal{V}}^{-T} - \mathbf{D}T_{\mathcal{V}}^{-1} (\mathbf{D}T_{\mathcal{V}}^{-T} \mathbf{D}\mathcal{W}^T + \mathbf{D}\mathcal{W} \mathbf{D}T_{\mathcal{V}}^{-1}) \mathbf{D}T_{\mathcal{V}}^{-T}), \\ \partial_{\mathcal{W}}(\det \mathbf{D}T_{\mathcal{V}}) &:= \det(\mathbf{D}T_{\mathcal{V}}) \operatorname{tr}(\mathbf{D}T_{\mathcal{V}}^{-1} \mathbf{D}\mathcal{W}), \end{aligned}$$

and where $p \in H_0^1(\Omega_0)$ is the solution to the adjoint problem

$$\begin{cases} -\operatorname{div}(\mathbf{M}_{\mathcal{V}} \operatorname{grad} p) = -j'(u)(\det \mathbf{D}T_{\mathcal{V}}) & \text{in } \Omega_0, \\ p = 0 & \text{on } \partial\Omega_0. \end{cases}$$

As Example 4.2 clearly illustrates, the Fréchet derivative of PDE constrained functionals depends on the solution u of the state problem and, possibly, on the solution p of the adjoint problem. As explicit analytic solutions of these boundary value problems are usually not available, one can replace them with approximate solutions, at the cost of introducing a perturbation error when solving the first order optimality condition (3.5). In particular, this perturbation error affects the quality of the descent directions.

We consider here approximations by means of the finite element method. When stated as a volume integral, the map $u \mapsto d\tilde{\jmath}(\mathcal{V}, u; \mathcal{W})$ is usually continuous with respect to the energy norm of u. Therefore, relying on standard duality techniques, one can expect to observe superconvergence in the approximation of the operator $\tilde{\jmath}(\mathcal{V}, u; \cdot)$ when the solution u is replaced by its finite element counterpart u_h. The same holds for evaluating the shape functional $\tilde{\jmath}(\mathcal{V}, u)$. In particular, we consider linear Lagrangian finite elements on quasi-uniform triangular meshes. In this case, it can be shown that [23, Theorem 3.1]

$$|d\tilde{\jmath}(\mathcal{V}, u; \mathcal{W}) - d\tilde{\jmath}(\mathcal{V}, u_h; \mathcal{W})| = C(\mathcal{V}) h^2 \|\mathcal{W}\|_{W^{2,4}(\mathbb{R}^d; \mathbb{R}^d)}, \tag{4.2}$$

Algorithm 1. Projected gradient method with Armijo rule

1: Select initial design Ω_0, optimization step δ, and parameters $\varepsilon,\gamma \in (0,1)$
2: Initialize $\mathcal{V}_N = 0$ and $\mathcal{V}_N^{\text{temp}} = 0$
3: Precompute all $\mathbf{B}_i$'s on quadrature nodes
4: **for** $ii = 1, \dots, MAX^{\text{ITER}}$ **do**
5: Assemble finite element stiffness matrix $\tilde{A}_{\mathcal{V}_N^{\text{temp}}}$ and load vector $\tilde{f}_{\mathcal{V}_N^{\text{temp}}}$
6: Compute finite element solution u_h of $\tilde{A}_{\mathcal{V}_N^{\text{temp}}} u = \tilde{f}_{\mathcal{V}_N^{\text{temp}}}$ in Ω_0
7: Compute $\tilde{\mathcal{J}}^{\text{new}} := \tilde{\mathcal{J}}(\mathcal{V}_N^{\text{temp}}, u_h)$
8: **if** $ii > 1$ **and** $\tilde{\mathcal{J}}^{\text{new}} - \tilde{\mathcal{J}}^{\text{old}} > \gamma\delta d\tilde{\mathcal{J}}(\mathcal{V}_N, u_h; \mathcal{V}_N^{\text{new}})$ **then**
9: Update $\delta \leftarrow \delta/2$
10: **else**
11: Update $\delta \leftarrow 2\delta$
12: Update $\mathcal{V}_N \leftarrow \mathcal{V}_N^{\text{temp}}$, $\tilde{\mathcal{J}}^{\text{old}} \leftarrow \tilde{\mathcal{J}}^{\text{new}}$
13: Compute $\mathcal{V}_N^{\text{new}} = \operatorname{argmin}_{\mathcal{W}_N \in V_N, \|\mathcal{W}_N\|_{H^1(D)}=1} d\tilde{\mathcal{J}}(\mathcal{V}_N, u_h; \mathcal{W}_N)$
14: **end if**
15: Set $\mathcal{V}_N^{\text{temp}} := \mathcal{V}_N + \delta\mathcal{V}_N^{\text{new}}$
16: **while** $\min(\det \mathbf{D}T_{\mathcal{V}_N^{\text{temp}}}) < \varepsilon$ **do**
17: Update $\delta \leftarrow \delta/2$ and set $\mathcal{V}_N^{\text{temp}} := \mathcal{V}_N + \delta\mathcal{V}_N^{\text{new}}$
18: **end while**
19: **end for**

where h denotes the width of the finite element mesh and $C(\mathcal{V})$ is a constant that depends on $\mathcal{V}$ and Ω_0. On the other hand, we do not observe superconvergence in (4.2) when $d\tilde{\mathcal{J}}$ is recast as an integration on the boundary $\partial\Omega_0$. We refer to [23] for more details.

The approximate optimal solution $\mathcal{V}_N^*$ can be computed iteratively by adopting the "simulation-based optimization policy": the routines to compute the solution of the state problem and the Fréchet derivative are "embedded into an optimization loop" [21, Section 1.1]. In this work we consider a descent method with Armijo rule [22, Section 2.2.1.1] as illustrated in Algorithm 1. The optimization algorithm is kept simple on purpose to allow for benchmarking. The next paragraphs give a detailed description of the algorithm's steps.

In line 13 we compute the descent direction. Since the Fréchet derivative $d\tilde{\mathcal{J}}(\mathcal{V}, u; \cdot)$ belongs to the dual space of $C^2(\overline{D}; \mathbb{R}^d)$, its descent direction is usually defined as the solution of

$$\inf_{\|\mathcal{W}\|_{C^2(\overline{D};\mathbb{R}^d)}=1} d\tilde{\mathcal{J}}(\mathcal{V}, u; \mathcal{W}),$$

see [22, p. 103]. However, such a descent direction may not exist because the space $C^2(\overline{D}; \mathbb{R}^d)$ is not reflexive.

Employing knowledge on the shape Hessian is also not straightforward, because the second order Fréchet derivative $d^2\tilde{\mathcal{J}}$ cannot be expected to be coercive in the $C^2(\overline{D}; \mathbb{R}^d)$-norm. Indeed, for any vector field $\mathcal{W}$ tangential to Ω_0 as well as for vector fields with a compact support that does not intersect $\partial\Omega_0$, it holds

$$d\tilde{\mathcal{J}}(\mathcal{V}, u; \mathcal{W}) = 0 \quad \text{and} \quad d^2\tilde{\mathcal{J}}(\mathcal{V}, u; \mathcal{W}, \mathcal{W}) = 0.$$

However, in several situations, the shape Hessian is a positive bilinear form when evaluated on vector fields with non-zero normal component on $\partial\Omega_0$. For instance, this is the case for the shape functional defined in (3.2), see [15]. Moreover, the shape Hessian can be expected to be a continuous bilinear form with respect to the $H^s(\partial\Omega_0)$-norm of the normal component of the vector fields. The regularity s of the "energy space" $H^s(\partial\Omega_0)$ depends on the problem under consideration; cf. [16]. In the seminal works [13, 16] it has been shown that shape optimization problems admit strict local minima, also called stable minimizers, if the shape Hessian is also coercive with respect to the $H^s(\partial\Omega_0)$-norm of the normal component of the vector fields, that is,

$$d^2\tilde{\mathcal{J}}(\mathcal{V}, u; \mathcal{W}, \mathcal{W}) \geq C\|\mathcal{W} \cdot \mathbf{n}\|_{H^s(\partial\Omega_0)}, \tag{4.3}$$

where $\mathbf{n}$ is the normal vector field on $\partial\Omega_0$ and $C > 0$ is a constant independent of $\mathcal{W}$. Thus, coercivity in the $H^s(\partial\Omega_0)$-norm can be used as a criterion to distinguish between well- and ill-posed shape optimization problems [16]. Therefore, from a theoretical point of view, it is natural to consider the $H^s(\partial\Omega_0)$ representative of the Fréchet derivative, which is unique up to extensions into the domain D of its values on $\partial\Omega_0$ in the normal direction $\mathbf{n}$. Note that for ill-posed shape optimization problems this choice provides a regularization in the spirit of regularized sequential quadratic programming [11].

We consider here descent directions given as $H^1_0(D;\mathbb{R}^d)$-representatives of the Fréchet derivative, that is, solutions to

$$\min_{\|\mathcal{W}\|_{H^1(D;\mathbb{R}^d)}=1} d\tilde{J}(\mathcal{V}, u; \mathcal{W}). \tag{4.4}$$

Up to a scaling factor, this amounts to solving the linear variational problem

$$(\mathcal{V}_N^{\text{new}}, \mathcal{W}_N)_{H^1(D;\mathbb{R}^d)} = d\tilde{J}(\mathcal{V}, u; \mathcal{W}_N) \quad \text{for all } \mathcal{W}_N \in V_N, \tag{4.5}$$

which is equivalent to solving a discrete Laplacian with homogeneous Dirichlet boundary conditions on ∂D and a spline based Galerkin discretization.

By the continuity of the normal Dirichlet trace operator

$$H^1_0(D;\mathbb{R}^d) \to H^{1/2}(\partial\Omega_0), \quad \mathcal{W} \mapsto \mathcal{W}\cdot\mathbf{n}|_{\partial\Omega_0},$$

the $H^1_0(D;\mathbb{R}^d)$-representative of the Fréchet derivative is the unique $H^1_0(D;\mathbb{R}^d)$-extension of the normal values of the $H^{1/2}(\partial\Omega_0)$-representative. Note that this is the proper "energy space" for the shape optimization problems considered in Section 6; cf. [15, 16].

In line 16 of Algorithm 1 we check that the transformation $T_{\mathcal{V}}$ is indeed a diffeomorphism. As suggested in [7], we verify that the value of $\det \mathbf{D}T_{\mathcal{V}}$ is bigger than a threshold value[1]. This relaxes the restrictive $C^2(\overline{D};\mathbb{R}^d)$-norm condition in (2.4) but still guarantees that the algorithm is well-defined. If $\min(\det \mathbf{D}T_{\mathcal{V}})$ is too small, we reduce the optimization stepsize until $T_{\mathcal{V}}$ defines a feasible transformation. Note that the while loop terminates due to the continuity of the determinant.

Remark 4.3. It might nevertheless happen that the (continuous) optimal solution $\mathcal{V}^*$ lies on the boundary of $B^1_{1-\varepsilon}$, and that, however, the value of $\tilde{J}(\mathcal{V}^*)$ is not yet satisfactory for convergence purposes. For instance, this might be the case when the initial guess Ω_0 is poorly chosen. In this situation a remedy is to select the retrieved shape as initial guess, and to restart the algorithm. Practically, this can be done by either creating a new mesh of $T_{\mathcal{V}^*}(\Omega_0)$ or by replacing the transformation $T_{\mathcal{V}}$ with the composition $T_{\mathcal{V}} \circ T_{\mathcal{V}^*}$ (exploiting the fact that the composition of diffeomorphisms is again a diffeomorphism). This latter approach can be made computationally affordable by simply re-evaluating all $\mathbf{B}_i$'s on the mapped quadrature nodes; see the next paragraph on the computational complexity of Algorithm 1. Note also that Theorems 2.1 and 2.5 still hold as long as a finite number of compositions is considered.

Finally, in line 9 we guarantee the admissibility of the optimization step δ according to the Armijo rule [22, Section 2.2.1.1].

5 Implementation in Matlab

We give details for an efficient implementation of Algorithm 1 in MATLAB for a two-dimensional problem. The state problem is solved by piecewise linear Lagrangian finite elements on triangles. The code can easily be extended to higher order polynomials by updating the routines accordingly. Our implementation follows the lines of [19, Section 3].

1 In practice, this condition can be tested only on a finite number of points. In our implementation we evaluate $\det \mathbf{D}T_{\mathcal{V}}$ on the FE quadrature points.

Firstly, we note that for each iteration the computational cost of Algorithm 1 is mainly due to:

- assembling the linear system in line 5,
- solving the linear system in line 6,
- evaluating the shape functional in line 7,
- computing the descent direction in line 13 (see (4.5)),
- testing the feasibility of the new descent direction in line 16.

Except for solving the linear system, these steps require the evaluation of $\mathbf{D}\mathcal{V}_N$, and thus of all partial derivatives of the B-splines, in each quadrature point on the finite element mesh. Since function calls are generally expensive, we pre-evaluate all these partial derivatives in all quadrature points in line 3, and assemble two sparse matrices `VdX` and `VdY` of size $(\texttt{nQP}\cdot\texttt{nElements})\times\texttt{nBsplines}$, where `nQP` is the number of quadrature points for triangle, `nElements` is the number of triangles, and `nBsplines` is the number of B-splines $\mathbf{B}_i$ (which corresponds to N in (4.1)). In these two matrices we store the values of the partial derivatives $\partial_x\mathbf{B}_i$ and $\partial_y\mathbf{B}_i$, respectively. Then, the entries of the Jacobian $\mathbf{D}\mathcal{V}_N$ can be obtained by multiplying the matrices `VdX` and `VdY` with the vectors `cX` and `cY`, whose entries are the coefficients c_i^1 and c_i^2 of the expansion of $\mathcal{V}_N$ given in (4.1), respectively. This yields an effective speed-up of the computations at the cost of memory. Note that the number of non-zero entries of `VdX` and `VdY` is significantly less than $\texttt{nQP}\cdot\texttt{nElements}\cdot\texttt{nBsplines}$ because B-splines have compact support.

With the matrices `VdX` and `VdY` at our disposal, it is very simple to test the feasibility of the new descent direction in line 16. We conclude this section by discussing the assembly of the stiffness matrix in line 5, the computation of the solution of the related linear system, the evaluation of the shape functional in line 7, and the computation of the descent direction in line 13.

The weak formulation of the state constraint of (3.3) suggests that

$$(u, v) \mapsto \int_{\Omega_0} \nabla u \cdot \mathbf{M}\nabla v \,\mathrm{d}\boldsymbol{x}$$

is a good representative bilinear form for the state constraint in line 6. By and large, $\mathbf{M}$ is a non-constant positive defined diffusion matrix. However, as the gradient of piecewise linear functions is piecewise constant, the matrix function $\mathbf{M}$ can be replaced by its mean values in each triangle. Then, a fully vectorized matrix assembly of the stiffness matrix can easily be implemented based on the details given in [19, Section 3.4], where Funken et al. describe the assembly of the stiffness matrix arising from the bilinear form

$$(u, v) \mapsto \int_{\Omega_0} \nabla u \cdot \nabla v \,\mathrm{d}\boldsymbol{x}.$$

The following step is to solve the linear system to retrieve the finite element solution u_h. In [6], Antonietti et al. show that multigrid strategies can be successfully applied in the context of shape optimization with moving meshes. We believe that similar ideas can be employed in our strategy. However, not to introduce additional error terms, we rely on the MATLAB function `mldivide`, which implements a direct solver.

Let $u^{\mathrm{ref}} \in H^1(\mathbb{R}^d)$. We consider

$$\mathcal{J} := \int_{\Omega} \nabla(u - u^{\mathrm{ref}}) \cdot \nabla(u - u^{\mathrm{ref}}) + (u - u^{\mathrm{ref}})^2 \,\mathrm{d}\boldsymbol{x},$$

as a representative for general shape functionals. The corresponding functional in parametric form reads

$$\tilde{\mathcal{J}} := \int_{\Omega_0} \nabla(u - \tilde{u}^{\mathrm{ref}}) \cdot \mathbf{M}_{\mathcal{V}_N}\nabla(u - \tilde{u}^{\mathrm{ref}}) + (u - \tilde{u}^{\mathrm{ref}})^2 \det(\mathbf{D}T_{\mathcal{V}_N}) \,\mathrm{d}\boldsymbol{x},$$

where $\tilde{u}^{\mathrm{ref}} := u^{\mathrm{ref}} \circ T_{\mathcal{V}_N}$. The contribution of the first integrand can be evaluated efficiently employing the stiffness matrix assembled in line 6. The second integrand is a scalar function, and does not represent any computational challenge, because the term $\det(\mathbf{D}T_{\mathcal{V}_N})$ can be computed efficiently with the matrices `VdX` and `VdY`.

The formulas of shape gradients of PDE constraint shape functionals strongly depend both on the shape functional itself and on the PDE constraint. Therefore, it is not possible to give a detailed description of its efficient implementation. We simply remark that, to compute the descent direction in line 13, the shape gradient has to be evaluated on all basis vector fields $\mathbf{B}_i\mathbf{e}_j$. By and large, the evaluation of the shape gradient on a fixed direction corresponds to an integration in volume, when the solution of the state problem is approximated with FEM [23]. Thus, $2\cdot$`nBsplines` integrals have to be computed. Employing MATLAB's pointwise arithmetics [19], these integrations can be performed simultaneously with a fully vectorized implementation. However, we stress that this step requires large amount of memory. We strongly recommend to exploit sparsity to reduce the active memory requirements. We finally recall that, in MATLAB, function input variables are not copied as long as they are not modified within the body of the function. We refer to the MATLAB documentation for further details.

6 Numerical Experiments

Let Ω_0 be an annular domain with internal boundary $\partial\Omega^{\text{in}}$ and external boundary $\partial\Omega^{\text{out}}$. The set of admissible domains is redefined to comprise domains obtained by perturbing only the external boundary $\partial\Omega^{\text{out}}$, i.e.,

$$\mathcal{U}_{\text{ad}}(\Omega_0) := \{T_{\mathcal{V}}(\Omega_0);\ T_{\mathcal{V}} = \mathcal{I} + \mathcal{V},\ \|\mathcal{V}\|_{C^2(\overline{D};\mathbb{R}^d)} \le 1-\varepsilon,\ \operatorname{supp}\mathcal{V}\cap\partial\Omega^{\text{in}} = \emptyset\}.$$

We consider the shape optimization problem

$$\inf_{\Omega\in\mathcal{U}_{\text{ad}}(\Omega_0)} \int_\Omega (\nabla u)^2 + g^2\,\mathrm{d}\boldsymbol{x} \quad \text{subject to} \quad \begin{cases} -\Delta u = 0 & \text{in } \Omega,\\ u = 0 & \text{on } \partial\Omega^{\text{out}},\\ u = 1 & \text{on } \partial\Omega^{\text{in}}, \end{cases} \tag{6.1}$$

where g is a constant.

Such an optimization problem belongs to the class of Bernoulli exterior free boundary problems, which are used as a benchmark in shape optimization because they admit stable minimizers. This is due to the $H^{1/2}(\partial\Omega_0)$-coercivity (see (4.3)) of its Hessian in the optimal shape [16].

The parametric form of (6.1) reads

$$\inf_{\mathcal{V}\in B^2_{1-\varepsilon}} \int_{\Omega_0} \nabla u\cdot\mathbf{M}_{\mathcal{V}}\nabla u + g^2\det\mathbf{D}T_{\mathcal{V}}\,\mathrm{d}\boldsymbol{x} \quad \text{subject to} \quad \begin{cases} -\operatorname{div}\mathbf{M}_{\mathcal{V}}\operatorname{grad} u = 0 & \text{in } \Omega_0,\\ u = 0 & \text{on } \partial\Omega_0^{\text{out}},\\ u = 1 & \text{on } \partial\Omega_0^{\text{in}}, \end{cases} \tag{6.2}$$

where $\mathbf{M}_{\mathcal{V}} := (\det\mathbf{D}T_{\mathcal{V}})\mathbf{D}T_{\mathcal{V}}^{-1}\mathbf{D}T_{\mathcal{V}}^{-T}$. The Fréchet derivative of the shape functional in (6.2) reads

$$d\tilde{\mathcal{J}}(\mathcal{V},u;\mathcal{W}) = \int_{\Omega_0} \nabla u\cdot(\partial_{\mathcal{W}}\mathbf{M}_{\mathcal{V}})\nabla u + g^2(\partial_{\mathcal{W}}\det\mathbf{D}T_{\mathcal{V}})\,\mathrm{d}\boldsymbol{x}. \tag{6.3}$$

Note that, in contrast to Example 4.2, formula (6.3) does not involve the solution of an adjoint problem [17].

Henceforth, $\partial\Omega_0^{\text{in}}$ is a circle of radius 0.5 centered in the origin. We set $g = (1.2\ \log(2.4))^{-1}$, so that the external boundary of the optimal solution is a circle of radius 1.2 centered in the origin. By rotational symmetry of the optimal solution, the minimal value of (6.1) is given by

$$\tilde{\mathcal{J}}^{\min} = 2\pi\Big(g^2\frac{1.2^2-0.5^2}{2} - \frac{1}{\log(0.5/1.2)}\Big).$$

In all the experiments, we consider finite element solutions computed with linear Lagrangian finite elements on quasi-uniform triangular meshes. Integrals in the domain are computed by a 3-point quadrature rule of order 3 in each triangle. The boundary of the computational domain is approximated by a polygon, which will not affect the convergence of linear finite elements [10, Section 10.2]. The optimization step δ is

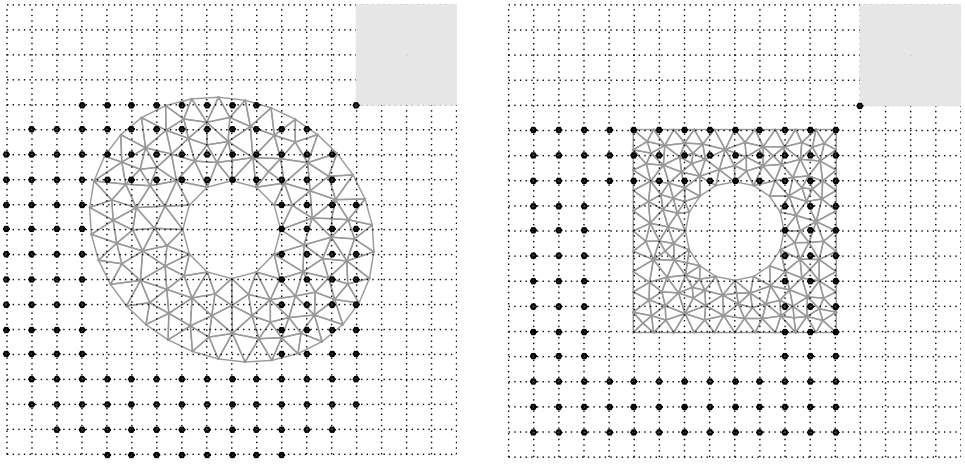

Figure 1. The initial guess Ω_0 is covered with a regular grid used to generate cubic B-splines. Dots indicate the lower left corner of the support of the active B-splines. The square in the top right corner indicates the support of a cubic B-spline. A triangular grid is generated on Ω_0 to compute the finite element solution u_h.

initially set to $\delta = 0.3$ and the parameter ε to $\varepsilon = 0.05$. Finally, we replace the Armijo rule condition

$$\tilde{\jmath}^{\text{new}} - \tilde{\jmath}^{\text{old}} \leq \gamma\delta d\tilde{\jmath}(\mathcal{V}_N, u_h; \mathcal{V}_N^{\text{new}})$$

with

$$|\tilde{\jmath}^{\text{new}} - \tilde{\jmath}^{\min}| - |\tilde{\jmath}^{\text{old}} - \tilde{\jmath}^{\min}| \leq \gamma\delta d\tilde{\jmath}(\mathcal{V}_N, u_h; \mathcal{V}_N^{\text{new}}),$$

so that the algorithm does not get stuck if, due to numerical error, $\tilde{\jmath}^{\text{new}}$ becomes smaller than the minimal value $\tilde{\jmath}^{\min}$. The parameter γ is set to $\gamma = 0.1$.

To show that the algorithm proposed in Section 4 is feasible, we select $\partial\Omega_0^{\text{out}}$ to be an ellipse with major semi-axis of length 1.5 and minor semi-axis of length 1.3; see Figure 1 (left). The domain Ω_0 is covered with a regular grid of width 0.255 over which the trial space V_N is constructed. The finite element solution u_h is computed on the mesh displayed in Figure 1 (left). Despite the coarseness of the mesh and the low resolution of the B-spline grid, after twelve optimization steps we already recover a satisfactory approximation of the target boundary; see Figure 2 (top left). The quality of the recovered solution improves if the finite element solution u_h is computed on a finer the mesh. The results obtained after three uniform refinements[2] of the mesh are displayed in Figure 2 (bottom left).

The experiment is repeated for a different initial design: a square with edges of length 2.06; see Figure 1 (right). Again, after twelve steps we recover a satisfactory approximation of the target boundary; see Figure 2 (top right). Although corners can not be smoothed with a diffeomorphism, the quality of the approximate solution improves by computing the finite element solution u_h on a finer mesh; see Figure 2 (bottom right).

Next, we investigate the impact of the finite element approximation on the retrieved approximate optimal solution. We keep the trial space of B-splines V_N fixed (with width 0.255), and we generate seven additional meshes through uniform refinement of the one displayed in Figure 1 (left). Let

$$\text{err}^{(i)} := \frac{|\tilde{\jmath}(\mathcal{V}_N^{(i)}, u_h) - \tilde{\jmath}_{\min}|}{\tilde{\jmath}(\mathcal{I}, u_h)} \tag{6.4}$$

be the scaled absolute error obtained after i steps of Algorithm 1. In Figure 3 (left) we plot the evolution of $\text{err}^{(i)}$ for each mesh. In Figure 3 (right) we plot $\text{err}^{(100)}$ for each mesh versus its mesh width. We observe an algebraic convergence with rate 1.7. We remark that $\tilde{\jmath}(\mathcal{V}, u_h)$ itself converges quadratically in the mesh width h (uniformly in $\mathcal{V} \in C^2(\overline{D}; \mathbb{R}^d)$).

2 During all mesh refinements the boundary nodes are projected onto $\partial\Omega_0$.

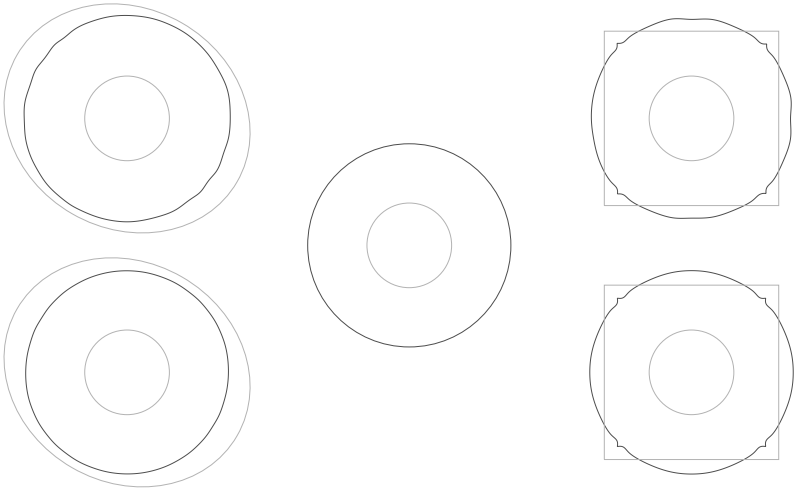

Figure 2. Approximate optimal boundary retrieved after twelve iterations of Algorithm 1. Light gray lines indicate the boundary of the initial guess Ω_0. Results in the first row are obtained using the meshes displayed in Figure 1. Despite the coarseness of the mesh and the low resolution of the B-spline grid, we recover a decent approximation (dark gray line) of the optimum (exterior boundary of the annulus in the middle). The results can be improved by computing on finer meshes (second row).

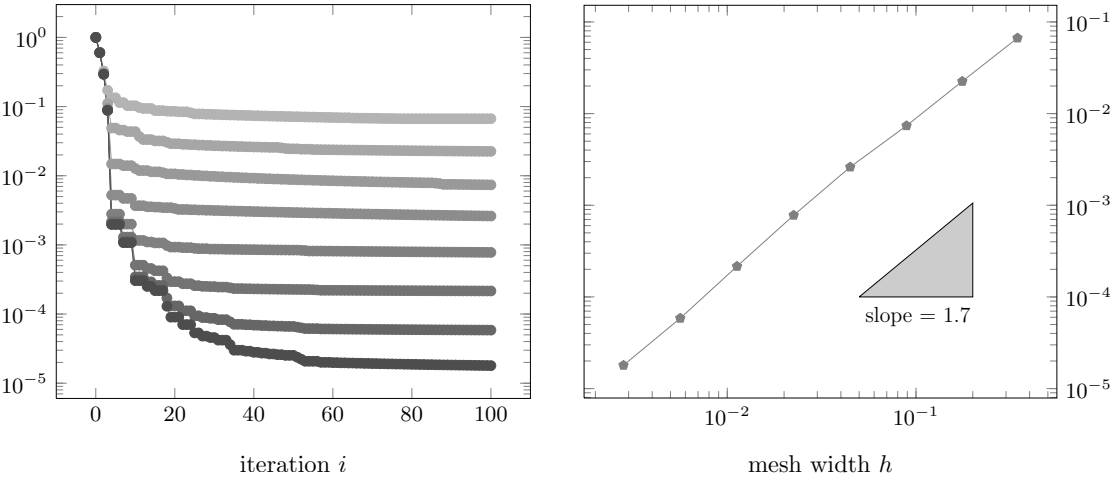

Figure 3. *Left:* Evolution of the scaled absolute error (6.4) on 8 nested meshes obtained with uniform refinement. *Right:* Value of scaled absolute error versus meshwidth at 100th iteration. We observe algebraic convergence with rate 1.7 (superconvergence).

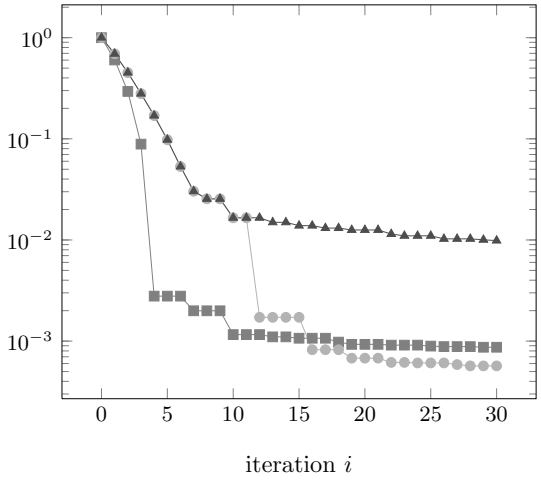

Figure 4. Evolution of the scaled absolute error (6.4) for a coarse (—▲—) and a finer (—■—) trial space V_N. Switching to a finer space after 10 iterations, it is possible to start with a coarse trial space and still retrieve an approximate solution with good quality (—●—).

Then, we investigate the impact of the resolution provided by V_N on the approximate optimal solution. We perform the experiment on the fifth mesh of the previous experiment. In Figure 4 we show the evolution of $\mathrm{err}^{(i)}$ for V_N constructed on a regular grid of width 0.51 (—▲—) and 0.255 (—■—). The former trial space comprises 54 active[3] B-splines, whilst the latter has 152 active B-splines. We see that the resolution of V_N affects the quality of the retrieved approximate optimal solution.

In real applications the exact optimal solution is usually not known a priori, and the minimum of the constrained shape functional might be bigger than zero. To investigate whether the algorithm has fully converged, we suggest to pursue an adaptive strategy by starting with a relatively coarse resolution and, when the iteration stagnates, to embed the so far computed discrete vector field on a nested space spanned by basis functions generated on a grid with half the meshwidth [24, Section 7.6]. Then, new descent directions are computed by taking into account only the new basis functions that intersect the boundary, whilst the ones that do not intersect the boundary are kept to provide a smoother decay of the vector field. The evolution of $\mathrm{err}^{(i)}$ for this strategy is displayed in Figure 4 (—●—). We see that we are able to improve the quality of the approximate optimal solution by switching to a finer space after ten iterations.

Finally, we test our algorithm on a prototypical ill-posed inverse problem. Let B be a fixed subdomain of a domain Ω and let $u_t \in L^2(B)$ be a given target function. The goal is to find the optimal domain that contains B, so that the shape functional

$$\mathcal{J}(\Omega) := \int_B (u - u_t)^2 \,\mathrm{d}\boldsymbol{x} \quad \text{subject to} \quad \begin{cases} -\Delta u = 1 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{6.5}$$

attains its minimum.

As explained in [12], elliptic regularity theory implies that the solution u of the state problem is in $H^2(\Omega)$ as soon as Ω is of class C^2. Therefore, the range of the operator $\mathcal{V} \mapsto u|_B$ is at most a dense subset of $L^2(\Omega)$ [9, Theorem 7.2]. Thus, the shape optimization problem (6.5) is ill-posed. An alternative explanation of the ill-posedness of (6.5) from a shape optimization point of view can be found in [15].

Similar to Example 4.2, the shape derivative of the shape optimization problem (6.5) recast in parametric form reads

$$d\tilde{\mathcal{J}}(\mathcal{V}, u; \mathcal{W}) = \int_{\Omega_0} \operatorname{grad} p \cdot \partial_{\mathcal{W}} \mathbf{M}_{\mathcal{V}} \operatorname{grad} u - p\, \partial_{\mathcal{W}} (\det \mathbf{D} T_{\mathcal{V}}) \,\mathrm{d}\boldsymbol{x},$$

where u and p are the solutions in $H^1_0(\Omega_0)$ of the weak form of

$$-\operatorname{div}(\mathbf{M}_{\mathcal{V}} \operatorname{grad} u) = \det \mathbf{D} T_{\mathcal{V}}, \quad -\operatorname{div}(\mathbf{M}_{\mathcal{V}} \operatorname{grad} p) = -\chi_B 2(u - u_t).$$

Note that here the set of admissible shapes reads

$$\mathcal{U}_{\mathrm{ad}}(\Omega_0) := \{T_{\mathcal{V}}(\Omega_0);\ T_{\mathcal{V}} = \mathcal{I} + \mathcal{V},\ \|\mathcal{V}\|_{C^2(\overline{D};\mathbb{R}^d)} \le 1 - \varepsilon,\ \operatorname{supp} \mathcal{V} \cap B = \emptyset\}.$$

In particular, we consider only vector fields that vanish on B because the latter denotes the region of interest and is assumed to be fixed.

The goal of this experiment is to asses the relevance of the regularization provided by the choice of $H^1_0(D;\mathbb{R}^d)$-representatives of the Fréchet derivative; see (4.4). We set $u_t(\boldsymbol{x}) := (1.2)^2/4 - \boldsymbol{x} \cdot \boldsymbol{x}/4$, so that an optimal domain is the disc centered in 0 with radius 1.2. The region of interest B is a disc centered in 0 with radius 0.5 whilst the initial domain Ω_0 is a disc centered in 0 with radius 1.3. We decide to start with Ω_0 close to the optimum because we construct the B-splines on a very fine grid in order to exclude regularization by discretization. To be precise, we set the gridwidth to 0.051, which corresponds to 729 active B-splines, and thus to 1458 basis vector fields; see (4.1). The finite element mesh has 126 465 nodes and 251 904 triangles.

In Figure 5 we display the evolution of the shape functional J (—▲—), the minimal value of $\det \mathbf{D} T_{\mathcal{V}}$ on the quadrature points (—■—), the optimization step δ (—●—). The graph on the left refers to the $H^1_0(D;\mathbb{R}^d)$-

3 We just consider B-splines whose support intersects $\partial\Omega^{\mathrm{out}}$.

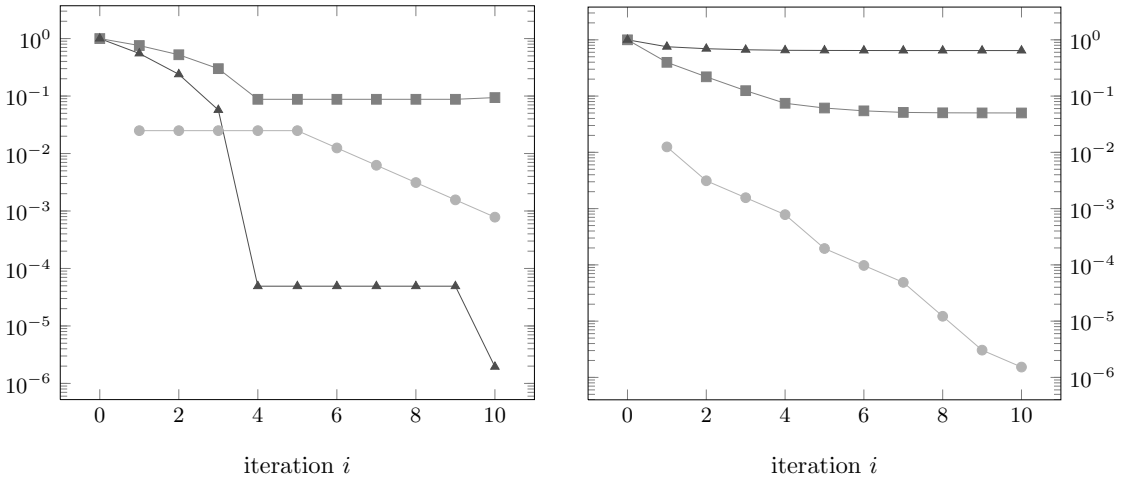

Figure 5. Evolution for the ill-posed shape optimization problem (6.5) of the shape functional $\mathcal{J}$ (—▲—), the minimal value of $\det(\mathbf{D}T_{\mathcal{V}})$ on the quadrature points (—■—), and of the optimization step δ (—●—) for descent directions computed with respect to the $H_0^1(\overline{D};\mathbb{R}^d)$- (*left*) and the $L^2(D;\mathbb{R}^d)$-metric (*right*). Due to the ill-posed nature of the shape optimization problem, the optimization step δ has to decrease rapidly to make the transformations $T_{\mathcal{V}}$ feasible in absence of regularization.

representative whilst the one on the right to the $L^2(D;\mathbb{R}^d)$-representative. For a better comparison, we normalize the descent directions with respect to the mean normal displacement

$$\frac{\int_{\partial\Omega_0} \mathcal{V}\cdot\mathbf{n}\,\mathrm{d}S}{\int_{\partial\Omega_0}\mathrm{d}S}.$$

When the $H^1(D;\mathbb{R}^d)$-metric is employed, we clearly see that the algorithm succeeds in reconstructing the target shape (small values of $\mathcal{J}$) and that the descent directions give rise to feasible transformations ($\min(\det\mathbf{D}T_{\mathcal{V}})$ is bigger than the threshold $\varepsilon = 0.05$) without making the optimization step δ decay rapidly to 0. Updates on δ occur only to fulfill the Armijo condition. On the other hand, the optimization step δ has to decrease rapidly to make the transformations $T_{\mathcal{V}}$ feasible when the algorithm relies on $L^2(D;\mathbb{R}^d)$-representatives of the Fréchet derivative. This is due to a steeper decay in the radial component of the retrieved vector field; see Figure 6. We also observe that descent directions computed in the $L^2(D;\mathbb{R}^d)$-metric are more oscillatory; see Figure 7. These facts drastically slow down the reconstruction of the optimal shape, and corroborate the regularizing properties provided by the use of the $H^1(D;\mathbb{R}^d)$-metric.

7 Conclusions

We presented a method to compute approximate optimal solutions of elliptic PDE constrained shape optimization problems. Shapes are identified with diffeomorphisms and the shape optimization problem is recast as an optimal control problem. The latter is then stated on a finite dimensional trial space based on cubic B-splines pursuing a Ritz approach. Under reasonable assumptions, the solution of the finite dimensional problem converges to the solution of the original problem.

To solve the finite dimensional problem we rely on descent methods. We employ $H_0^1(D;\mathbb{R}^d)$-representatives of the Fréchet derivative. For the well-posed shape optimization problem (6.1), this choice is consistent with the coercivity estimate (4.3) fulfilled by the shape Hessian. For the ill-posed problem (6.5), it provides a regularization in the spirit of regularized sequential quadratic programming.

Superconvergence in the approximation of the Fréchet derivative can be achieved relying on FE discretizations of the underlying BVP. Numerical experiments show that accuracy in the approximation of the Fréchet derivative directly affects the quality of the retrieved approximate optimal solution.

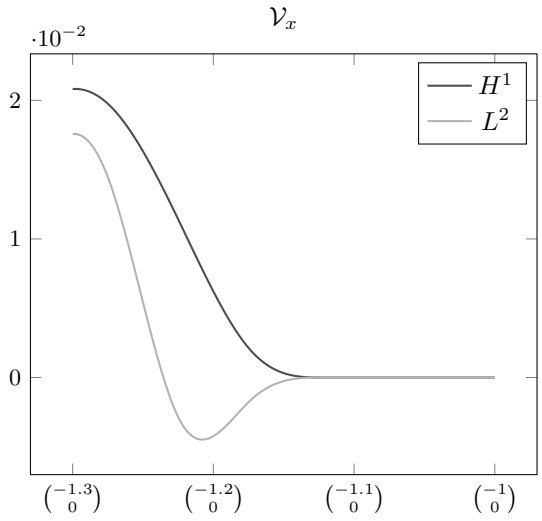

Figure 6. Radial component of the descent direction $\delta\mathcal{V}_N^{\text{new}}$ (with $\delta = 0.025$) computed in the first optimization step in the $H^1(D;\mathbb{R}^d)$- (*dark gray*) and the $L^2(D;\mathbb{R}^d)$-metric (*light gray*) on the path from $(-1.3, 0)$ to $(-1, 0)$. The descent direction computed in the $L^2(D;\mathbb{R}^d)$-metric suffers from a steeper decay in the radial component.

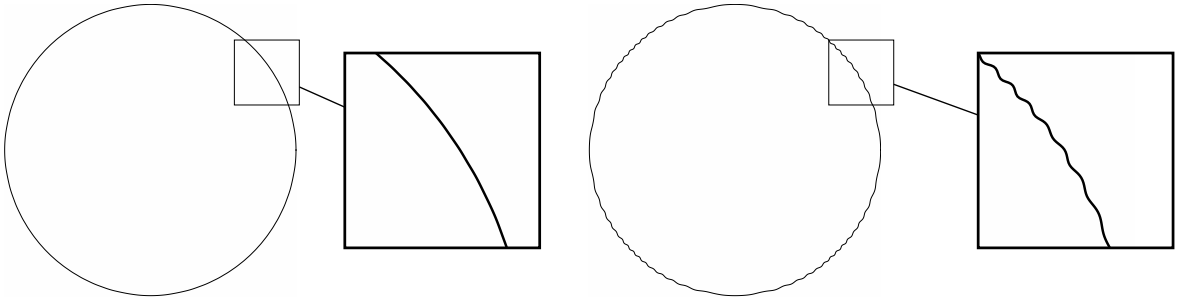

Figure 7. Shapes $T_{\mathcal{V}}(\Omega_0)$ obtained using the descent direction $\delta\mathcal{V}_N^{\text{new}}$ (with $\delta = 0.025$) computed in the first optimization step in the $H^1(D;\mathbb{R}^d)$- (*left*) and the $L^2(D;\mathbb{R}^d)$-metric (*right*). In the latter case we observe high-frequency oscillations.

Finally, we discussed an adaptive strategy based on nested trial spaces to balance discretization errors due to B-splines approximation of shapes and FE approximations of the solution of the PDE constraint.

Acknowledgment: We thank Andreas Hiltebrand for fruitful discussions on implementation aspects.

Funding: The work of Alberto Paganini was partly supported by ETH Grant CH1-02 11-1. He acknowledges RICAM fellowship for the Special Semester on Calculus of Variations, during which part of this work was completed.

References

[1] R. A. Adams and J. J. F. Fournier, *Sobolev Spaces*, 2nd ed., Elsevier/Academic Press, Amsterdam, 2003.
[2] G. Allaire, *Conception optimale de structures*, Springer, Berlin, 2007.
[3] G. Allaire, C. Dapogny and P. Frey, Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh, *C. R. Math. Acad. Sci. Paris* **349** (2011), no. 17-18, 999–1003.
[4] G. Allaire, C. Dapogny and P. Frey, Shape optimization with a level set based mesh evolution method, *Comput. Methods Appl. Mech. Engrg.* **282** (2014), 22–53.
[5] G. Allaire, F. Jouve and A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comput. Phys.* **194** (2004), no. 1, 363–393.
[6] P. F. Antonietti, A. Borzì and M. Verani, Multigrid shape optimization governed by elliptic PDEs, *SIAM J. Control Optim.* **51** (2013), no. 2, 1417–1440.
[7] F. Ballarin, A. Manzoni, G. Rozza and S. Salsa, Shape optimization by free-form deformation: Existence results and numerical solution for Stokes flows, *J. Sci. Comput.* **60** (2014), no. 3, 537–563.

[8] D. Begis and R. Glowinski, Application de la méthode des éléments finis à l'approximation d'un problème de domaine optimal. Méthodes de résolution des problèmes approchés, *Appl. Math. Optim.* **2** (1975/76), no. 2, 130–169.
[9] D. Braess, *Finite Elements. Theory, Fast Solvers, and Applications in Elasticity Theory*, 3rd ed., Cambridge University Press, Cambridge, 2007.
[10] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Springer, New York, 2008.
[11] M. Burger and W. Mühlhuber, Iterative regularization of parameter identification problems by sequential quadratic programming methods, *Inverse Problems* **18** (2002), no. 4, 943–969.
[12] D. Chenais and E. Zuazua, Controllability of an elliptic equation and its finite difference approximation by the shape of the domain, *Numer. Math.* **95** (2003), no. 1, 63–99.
[13] M. Dambrine and M. Pierre, About stability of equilibrium shapes, *M2AN Math. Model. Numer. Anal.* **34** (2000), no. 4, 811–834.
[14] M. C. Delfour and J.-P. Zolésio, Velocity method and Lagrangian formulation for the computation of the shape Hessian, *SIAM J. Control Optim.* **29** (1991), no. 6, 1414–1442.
[15] K. Eppler and H. Harbrecht, Coupling of FEM and BEM in shape optimization, *Numer. Math.* **104** (2006), no. 1, 47–68.
[16] K. Eppler and H. Harbrecht, Shape optimization for free boundary problems. Analysis and numerics, in: *Constrained Optimization and Optimal Control for Partial Differential Equations*, Internat. Ser. Numer. Math. 160, Birkhäuser, Basel (2012), 277–288.
[17] K. Eppler, H. Harbrecht and R. Schneider, On convergence in elliptic shape optimization, *SIAM J. Control Optim.* **46** (2007), no. 1, 61–83.
[18] I. Fumagalli, N. Parolini and M. Verani, Shape optimization for Stokes flow: A reference domain approach, preprint (2014), http://arxiv.org/abs/1403.3540.
[19] S. Funken, D. Praetorius and P. Wissgott, Efficient implementation of adaptive P1-FEM in Matlab, *Comput. Methods Appl. Math.* **11** (2011), no. 4, 460–490.
[20] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*, Prentice-Hall, Englewood Cliffs, 1963.
[21] R. Herzog and K. Kunisch, Algorithms for PDE-constrained optimization, *GAMM-Mitt.* **33** (2010), no. 2, 163–176.
[22] M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich, *Optimization with PDE Constraints*, Springer, New York, 2009.
[23] R. Hiptmair, A. Paganini and S. Sargheini, Comparison of approximate shape gradients, *BIT* **55** (2015), 459–485.
[24] K. Höllig and J. Hörner, *Approximation and Modeling with B-Splines*, Society for Industrial and Applied Mathematics, Philadelphia, 2013.
[25] B. Kiniger and B. Vexler, A priori error estimates for finite element discretizations of a shape optimization problem, *ESAIM Math. Model. Numer. Anal.* **47** (2013), no. 6, 1733–1763.
[26] T. Lassila and G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, *Comput. Methods Appl. Mech. Engrg.* **199** (2010), no. 23-24, 1583–1592.
[27] P. Morice, Une methode d'optimisation de forme de domaine, in: *Control Theory, Numerical Methods and Computer Systems Modelling*, Springer, Berlin (1975), 454–467.
[28] O. Pironneau, Optimal shape design for elliptic systems, in: *System Modeling and Optimization*, Springer, Berlin (1982), 42–66.
[29] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, 1999.
[30] M. Souli and J.-P. Zolésio, Shape derivative of discretized problems, *Comput. Methods Appl. Mech. Engrg.* **108** (1993), no. 3–4, 187–199.