

# FIRST AND SECOND ORDER SHAPE OPTIMIZATION BASED ON RESTRICTED MESH DEFORMATIONS\*

TOMMY ETLING<sup>†</sup>, ROLAND HERZOG<sup>†</sup>, ESTEFANÍA LOAYZA<sup>†</sup>, AND GERD WACHSMUTH<sup>‡</sup>

**Abstract.** We consider shape optimization problems subject to elliptic partial differential equations. In the context of the finite element method, the geometry to be optimized is represented by the computational mesh, and the optimization proceeds by repeatedly updating the mesh node positions. It is well known that such a procedure eventually may lead to a deterioration of mesh quality, or even an invalidation of the mesh, when interior nodes penetrate neighboring cells. We examine this phenomenon, which can be traced back to the ineptness of the discretized objective when considered over the space of mesh node positions. As a remedy, we propose a restriction in the admissible mesh deformations, inspired by the Hadamard structure theorem. First and second order methods are considered in this setting. Numerical results show that mesh degeneracy can be overcome, avoiding the need for remeshing or other strategies. FENICS code for the proposed methods is available on GitHub.

**Key words.** shape optimization, shape gradient descent, shape Newton method, restricted mesh deformations

**AMS subject classifications.** 90C30, 90C46, 65K05

**1. Introduction.** Shape optimization is ubiquitous in the design of structures of all kinds, going from drug eluting stents [Zunino, 2004](#) until aircraft wings [Schmidt, Schulz, et al., 2011](#) or horn-like structures appearing in devices for acoustic or electromagnetic waves [Udawalpola, Berggren, 2008](#). All of these and many other applications involve the solution  $u$  of a partial differential equation (PDE), so the general formulation of shape optimization problems considered here is as follows:

$$(1.1) \quad \min_{\Omega} j(\Omega, u(\Omega)).$$

Here  $u(\Omega)$  is the solution of the underlying PDE defined on the domain  $\Omega$ , which is to be optimized. In the following, we will mainly use the reduced objective  $J(\Omega) := j(\Omega, u(\Omega))$ .

Computational approaches to solving PDE-constrained shape optimization problems usually proceed along the following lines. First, one derives an expression for the *shape derivative* of the objective w.r.t. vector fields which describe the perturbation

---

\*Submitted to the editors July 12, 2019.

**Funding:** This work was partially supported by DFG grants HE 6077/10–1 and WA 3636/4–1 within the [Priority Program SPP 1962](#), which is gratefully acknowledged.

<sup>†</sup>Technische Universität Chemnitz, Faculty of Mathematics, Professorship Numerical Mathematics (Partial Differential Equations), 09107 Chemnitz, Germany (TE: [tommy.etling@tu-chemnitz.de](mailto:tommy.etling@tu-chemnitz.de), [https://www.tu-chemnitz.de/mathematik/part\\_dgl/people/etling](https://www.tu-chemnitz.de/mathematik/part_dgl/people/etling); RH: [roland.herzog@mathematik.tu-chemnitz.de](mailto:roland.herzog@mathematik.tu-chemnitz.de), <https://www.tu-chemnitz.de/herzog>; EL: [estefania.loayza@mathematik.tu-chemnitz.de](mailto:estefania.loayza@mathematik.tu-chemnitz.de), [https://www.tu-chemnitz.de/mathematik/part\\_dgl/people/loayza](https://www.tu-chemnitz.de/mathematik/part_dgl/people/loayza)).

<sup>‡</sup>Brandenburgische Technische Universität Cottbus-Senftenberg, Institute of Mathematics, Chair of Optimal Control, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany ([gerd.wachsmuth@b-tu.de](mailto:gerd.wachsmuth@b-tu.de), <https://www.b-tu.de/fg-optimale-steuerung>).

of the current domain  $\Omega$ . The perturbations are carried out either in terms of the perturbation of identity, or the velocity method. We refer the reader to [Delfour, Zolésio, 2011](#), Chapters 4.3–4.4 for details. The shape derivative can be stated either as an expression concentrated on the boundary  $\partial\Omega$ , or as a volume expression. The first is due to the Hadamard structure theorem ([Sokołowski, Zolésio, 1992](#), Theorem 2.27). For volume expressions, we refer the reader, for instance, to [Laurain, Sturm, 2016](#); [Hiptmair, Paganini, Sargheini, 2015](#). Second, the shape derivative, which represents a linear functional on the perturbation vector fields, needs to be converted into a vector field  $V$  itself, often referred to as the *shape gradient*. This can be achieved by evaluating the Riesz representative of the derivative w.r.t. an inner product. The latter is often chosen as the bilinear form associated with the Laplace-Beltrami operator on  $\partial\Omega$ , or with the linear elasticity (Lamé) system on  $\Omega$ , see e.g. [Schmidt, Schulz, et al., 2011](#); [Schulz, Siebenborn, 2016](#); [Schmidt, Schulz, 2010](#); [2009](#). More sophisticated techniques include quasi-Newton or Hessian-based inner products; see [Eppler, Harbrecht, 2005](#); [Novruzi, Roche, 2000](#); [Schulz, Siebenborn, Welker, 2015](#); [Schulz, 2014](#). This perturbation field is then used to update the domain  $\Omega$  inside a line search method, where the transformed domain

$$(1.2) \quad \Omega_\alpha = \{x + \alpha V(x) : x \in \Omega\}$$

associated with the step size  $\alpha$  is obtained from the perturbation of identity approach.

Using the boundary expression of the shape derivative provides an alternative to the domain transformation approach described in the previous paragraph. In this case, a normal vector field concentrated on the current domain boundary  $\partial\Omega$  is obtained, which provides a descent direction. In the presence of a PDE, modifications of the domain boundary need to be accompanied by a deformation strategy in the interior (see e.g. [Schmidt, Ilic, et al., 2011](#)). This two-stage process, however, causes discontinuities in the discrete objective function and thus disturbs the performance of optimization algorithms. On the other hand, the use of the volume expression has been shown to exhibit superior regularity and better finite element approximation results compared to the boundary expression (see e.g. [Hiptmair, Paganini, Sargheini, 2015](#)).

In any case, while the computation of the shape derivative is either based on the continuous or some discrete formulation of problem (1.1), the computation of the shape gradient and the subsequent updating steps will always be carried out in the discrete setting. Typically, the shape  $\Omega$  is represented by a computational mesh, and the underlying PDE is solved, e.g., by the finite element method. The perturbation field  $V$  is then expressed as a piecewise linear field, i.e., it is represented in terms of a velocity vector attached to each vertex position. The domain  $\Omega$  is subsequently updated according to (1.2) inside a line search procedure.

It has been observed in many publications that this straightforward approach has one major drawback: it often leads to a degeneracy of the computational mesh. This degeneracy manifests itself in different ways, but mostly through degrading cell aspect ratios, or even mesh nodes entering neighboring cells. [Doğan et al., 2007](#) for instance observe that such mesh distortions impair computations and lead to numerical artifacts. In practice, both phenomena often lead to a breakdown of computational shape optimization procedures.

In [section 4](#) we shed some light on this process of mesh destruction. We attribute it to a discretization artifact, by which the positions of *all* mesh nodes of a computational

mesh have an impact on the discrete solution of the PDE present in the problem. This presents optimization routines with an opportunity to shift the mesh nodes in such a way that the discrete solution of the PDE exhibits features which allow further descent in the objective, but at the expense of mesh quality and solution accuracy of the PDE. Notice that this issue does not arise in the continuous setting, where the redistribution of material points in the interior of the domain has no effect on the PDE solution and thus on the objective. In lack of a better name, we refer to the phenomenon described above as “spurious descent directions” since they are, indeed, leading us further away from the solution of the continuous problem.

Over the past 10 years, a range of various techniques have been proposed to circumvent this major obstacle in computational shape optimization. A natural choice is to remesh the computational domain; see for instance [Wilke, Kok, Groenwold, 2005](#); [Morin et al., 2012](#); [Sturm, 2016](#); [Dokken et al., 2018](#); [Feppon et al., 2018](#). Remeshing can be carried out either in every iteration or whenever some measure of mesh quality falls below a certain threshold. Drawbacks of remeshing include the high computational cost and the discontinuity introduced into the history of the objective values.

[Bänsch, Morin, Nochetto, 2005](#); [Doğan et al., 2007](#) describe several techniques such as mesh regularization, space adaptivity, angle control in addition to a semi-implicit Euler discretization for the velocity method, with time adaptivity and backtracking line search. In a follow-up work, [Morin et al., 2012](#) consider a line search method that aims to avoid mesh distortion due to tangential movements of the boundary nodes, combined with a geometrically consistent mesh modification (GCM) proposed in [Bonito, Nochetto, Pauletti, 2010](#). [Giacomini, Pantz, Trabelsi, 2017](#) address the issue of spurious descent directions, attributed to discretization errors in the underlying PDE model, via a goal-oriented mesh adaptation approach. Recently, [Iglesias, Sturm, Wechsung, 2017](#) proposed to enforce shape gradients from nearly conformal transformations, which are known to preserve angles and ensure a good quality of the mesh along the optimization process.

Finally, we mention [Schulz, Siebenborn, Welker, 2015](#); [2016](#); [Schulz, Siebenborn, 2016](#), who advocate the linear elasticity model as the inner product to convert shape derivative into a shape gradient. In particular in [Schulz, Siebenborn, Welker, 2016](#) the authors propose to omit the assembly of interior contributions appearing in the discrete volume expression of the shape derivative. This approach is related to but conceptionally different from our idea and no analysis is provided there. A thorough comparison is provided in [subsection 5.3](#).

**Our Contribution.** In this paper we propose an approach to avoid spurious descent directions in the course of numerical shape optimization procedures, which is different from all of the above and does not require remeshing. The main idea is based on the observation that—in the continuous setting—shape gradients are perturbation fields which are generated exclusively by normal forces on the boundary of the current domain. This follows from the Hadamard structure theorem. However, in the discrete setting, the Hadamard structure theorem is not available, and thus classical discrete shape gradients also contain contributions from interior forces and tangential boundary forces. We therefore propose to project the shape gradient onto the subspace of perturbation fields generated by normal forces. We refer to this approach as *restricted mesh deformations*.

We demonstrate that the proposed approach indeed avoids spurious descent directions and degenerate meshes. As a consequence, we can solve discrete shape optimization problems to high accuracy, i.e., very small norm of the restricted gradient. Both gradient and Newton schemes in 2D and 3D are considered. An implementation in the finite element software FENICS is available as open-source on GitHub; see [Etling et al., 2018](#).

The paper is structured as follows. In [section 2](#) we present a shape optimization model problem and prove, as an auxiliary result, the existence of a globally optimal domain. In [section 3](#) we review the volume and boundary representations of the shape derivative. In [section 4](#) we consider the discrete counterpart of the model problem and its shape derivative. We also illustrate the detrimental effect of spurious descent directions. The main idea of restricted mesh deformations is introduced in [section 5](#). An associated *restricted gradient scheme* is also introduced and its performance is compared to the classical shape gradient method in [section 6](#). [Sections 7](#) and [8](#) are devoted to second-order shape derivatives in the restricted setting and the demonstration of the associated Newton scheme. Conclusions are given in [section 9](#).

We wish to point out that the model problem considered throughout the paper is clearly academic. It was chosen since, as an auxiliary result of the present paper, we present a new technique to prove the existence of optimal shapes, which requires certain properties of the objective to hold; see [Remark 2.3](#). It should be understood that our main idea of considering restricted mesh deformations to avoid spurious descent directions applies to a much broader class of PDE constrained shape optimization problems.

**2. Preliminaries.** Throughout the paper, we consider the following model problem,

$$(2.1) \quad \text{Minimize} \quad \int_{\Omega} u \, dx \quad \text{s.t.} \quad \Omega \subset D \text{ is open, } \begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

Here the optimization variable  $\Omega \subset \mathbb{R}^d$  is an admissible domain contained in some bounded and open hold-all  $D \subset \mathbb{R}^d$ , and  $f \in H^1(D)$  is a given right hand side. The elliptic state equation is understood in weak form,

$$(2.2) \quad \text{Find } u \in H_0^1(\Omega) \quad \text{such that} \quad \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx \quad \forall v \in H_0^1(\Omega).$$

The next result shows that our shape optimization problem [\(2.1\)](#) has a solution if we slightly relax the class of admissible sets. We will see that it is sufficient to consider *quasi-open* rather than open sets. For an introduction of quasi-open sets, quasi-continuity, quasi-everywhere (q.e.) and related notions, we refer the reader to [Attouch, Buttazzo, Michaille, 2014](#), Section 5.8. We consider the slightly relaxed problem

$$(2.3) \quad \text{Minimize} \quad \int_{\Omega} u \, dx \quad \text{s.t.} \quad \Omega \subset D \text{ is quasi-open, } -\Delta u = f \text{ in } H^{-1}(\Omega).$$

Let us recall that  $H_0^1(\Omega) = \{u \in H_0^1(\mathbb{R}^d) \mid u = 0 \text{ q.e. in } \mathbb{R}^d \setminus \Omega\}$  and  $H^{-1}(\Omega)$  is the dual space of  $H_0^1(\Omega)$ . The PDE in [\(2.3\)](#) is also to be understood in the weak sense,

i.e.,

$$\text{Find } u \in H_0^1(\Omega) \quad \text{such that} \quad \int_D \nabla u \cdot \nabla v = \int_D f v \, dx \quad \forall v \in H_0^1(\Omega).$$

We emphasize that the main reason for this existence result is that the objective is monotone w.r.t. the state  $u$ , see also [Remark 2.3](#) below.

**THEOREM 2.1.** *Problem (2.3) admits a global minimizer  $(\hat{\Omega}, \hat{u})$ .*

Note that the extreme case  $(\hat{\Omega}, \hat{u}) = (\emptyset, 0)$  is possible.

*Proof.* First, we remark that it is sufficient to consider only pairs  $(\{u < 0\}, u)$  with  $u \leq 0$  in (2.3). Indeed, if  $(\Omega, u)$  is any admissible pair, we can consider  $(\{u < 0\}, \min(u, 0))$  in its stead. Note that  $\{u < 0\}$  is quasi-open since  $u$  can be chosen to be quasi-continuous. This pair is again admissible due to

$$\int_D \nabla \min(u, 0) \cdot \nabla v \, dx = \int_\Omega \nabla u \cdot \nabla v \, dx = \int_D f v \, dx \quad \forall v \in H_0^1(\{u < 0\}),$$

since  $v = 0$  q.e. on  $\Omega \setminus \{u < 0\}$ . Moreover, the objective value of  $(\{u < 0\}, \min(u, 0))$  is not larger than the objective value of  $(\Omega, u)$ .

Now, let  $\{(\Omega_n, u_n)\}$  be a minimizing sequence for (2.3) with  $u_n \leq 0$  and  $\Omega_n = \{u_n < 0\}$ . It is clear that the sequence  $\{u_n\}$  is bounded in  $H_0^1(D)$ , therefore we can extract a weakly convergent subsequence (without relabeling) with weak limit  $u$ . Clearly,  $u \leq 0$ . Now we define  $\hat{\Omega} = \{u < 0\}$  and denote by  $\hat{u} \in H_0^1(\hat{\Omega})$  the solution of  $-\Delta \hat{u} = f$  in  $H^{-1}(\hat{\Omega})$ . It remains to check that  $\hat{u} \leq u$  holds since this implies the global optimality of  $\hat{u}$  (due to the monotonicity of the objective). To this end, we choose an arbitrary  $v \in H_0^1(D)$  such that  $-u \geq v \geq 0$ . For  $v_n := \min(-u_n, v)$  we have  $v_n \in H_0^1(\Omega_n)$  due to  $v \geq 0$ . Moreover,  $v_n \rightharpoonup \min(-u, v) = v$  in  $H_0^1(D)$ , see [Wachsmuth, 2016](#), Lemma 4.1. Thus,

$$\begin{aligned} \int_D f v \, dx &= \lim_{n \rightarrow \infty} \int_D f v_n \, dx = \lim_{n \rightarrow \infty} \int_D \nabla u_n \cdot \nabla v_n \, dx \\ &= \lim_{n \rightarrow \infty} \int_D \nabla(u_n + v) \cdot \nabla(v_n - v) + \nabla u_n \cdot \nabla v - \nabla v \cdot \nabla(v_n - v) \, dx \\ &= \lim_{n \rightarrow \infty} \int_D -\nabla |\min(-u_n - v, 0)|^2 + \nabla u_n \cdot \nabla v \, dx \leq \int_D \nabla u \cdot \nabla v \, dx. \end{aligned}$$

Since  $v \in H_0^1(\hat{\Omega})$ , we can test the equation for  $\hat{u}$  with  $v$  and we find

$$\int_{\hat{\Omega}} \nabla(\hat{u} - u) \cdot \nabla v \, dx \leq 0 \quad \forall v \in H_0^1(D) \text{ satisfying } -u \geq v \geq 0.$$

Now, we can use a density argument, see [Mignot, 1976](#), Lemme 3.4, to obtain that this inequality holds for all  $v \in H_0^1(\hat{\Omega})$  which satisfy  $v \geq 0$ . Using  $v = \max(\hat{u} - u, 0)$  implies  $\max(\hat{u} - u, 0) = 0$ , i.e.,  $\hat{u} \leq u$ . Finally, the optimality of  $(\hat{\Omega}, \hat{u})$  follows from

$$\int_D \hat{u} \, dx \leq \int_D u \, dx = \lim_{n \rightarrow \infty} \int_D u_n \, dx. \quad \square$$

*Remark 2.2.* There is a deeper reason for  $\hat{u} \leq u$  being true in the above proof. Indeed, using the theory of relaxed Dirichlet problems, one can show that  $u$  satisfies  $-\Delta u + \mu u = f$  for some capacity measure  $\mu$ . We refer to [Attouch, Buttazzo, Michaille, 2014](#), Section 5.8.4 for a nice introduction to capacity measures. Due to  $u \leq 0$  we have (in a certain sense)  $\mu u \leq 0$  and therefore  $\hat{u} \leq u$  follows from the maximum principle since “ $-\Delta \hat{u} = f \leq f - \mu u = -\Delta u$ ”. However, we included the above direct proof because it does not rely on the notion of capacity measures.

*Remark 2.3.* The above proof of existence generalizes to a larger class of objective functionals. In fact, we can replace the objective in (2.3) with

$$\int_{\Omega} j(x, u(x)) \, dx$$

if the integrand  $j$  satisfies

$$(2.4a) \quad j(x, \cdot) \text{ is monotonically increasing on } (-\infty, 0] \text{ and non-negative on } [0, \infty),$$

$$(2.4b) \quad j(\cdot, u) \in L^1(D) \quad \forall u \in H_0^1(D),$$

$$(2.4c) \quad u_n \rightharpoonup u \text{ in } H_0^1(D) \text{ implies } \int_D j(u) \, dx \leq \liminf_{n \rightarrow \infty} \int_D j(u_n) \, dx.$$

Under these general assumptions, one can use the same proof as the one given for [Theorem 2.1](#) above, but the final estimate has to be replaced by

$$\begin{aligned} \int_{\hat{\Omega}} j(\cdot, \hat{u}) \, dx &\leq \int_{\hat{\Omega}} j(\cdot, u) \, dx = \int_D j(\cdot, u) - j(\cdot, 0) \, dx + \int_{\{u < 0\}} j(0) \, dx \\ &\leq \liminf_{n \rightarrow \infty} \int_D j(\cdot, u_n) - j(\cdot, 0) \, dx + \int_{\{u_n < 0\}} j(0) \, dx \\ &= \liminf_{n \rightarrow \infty} \int_{\Omega_n} j(\cdot, u_n) \, dx. \end{aligned}$$

Note that Fatou’s lemma together with  $u_n \rightarrow u$  a.e. (along a subsequence) implies

$$\int_{\{u < 0\}} j(0) \, dx \leq \liminf_{n \rightarrow \infty} \int_{\{u_n < 0\}} j(0) \, dx.$$

Again, this shows the optimality of  $(\hat{\Omega}, \hat{u})$ .

**3. Shape Calculus.** This section is devoted to the presentation of the shape differentiability of problem (2.1). Since this is rather standard problem we will be able to directly apply results from [Ito, Kunisch, Peichl, 2008](#). To this end, we assume that both the hold-all  $D \subset \mathbb{R}^d$  and  $\Omega \subset \mathbb{R}^d$  are open and have  $C^{1,1}$ -boundaries  $\partial D$  and  $\partial \Omega$ , respectively. Moreover we assume  $\bar{\Omega} \subset D$  so that  $\Omega$  has a positive distance to the boundary of  $D$ .

We are describing variations of the domain  $\Omega$  by the *perturbation of identity method*, i.e., we consider a family of transformations  $\{T_\alpha\}_{\alpha \in [0, \tau]}$  such that

$$(3.1) \quad T_\alpha = \text{id} + \alpha V,$$

where  $V \in C^{1,1}(D)^d$  is a given vector field. The family  $\{T_\alpha\}$  creates a family of perturbed domains  $\Omega_\alpha = T_\alpha(\Omega)$ . In view of Banach’s fixed point theorem, there exists a bound  $\tau > 0$  such that  $T_\alpha$  is invertible for all  $\alpha \in [0, \tau]$ .

By a straightforward application of [Ito, Kunisch, Peichl, 2008](#), Theorem 2.1 we obtain the following result.

**THEOREM 3.1.** *The shape functional given in (2.1) is shape differentiable and its shape derivative in the direction of the perturbation field  $V$  is given by*

$$(3.2) \quad J'(\Omega; V) = \int_{\Omega} u (\operatorname{div} V) \, dx + \int_{\Omega} (\nabla u)^{\top} [(\operatorname{div} V) \operatorname{id} - DV - DV^{\top}] \nabla p \, dx - \int_{\Omega} \operatorname{div}(f V) p \, dx$$

where  $DV$  denotes the Jacobian of  $V$  and the adjoint state  $p$  is the unique solution of the following adjoint problem,

$$(3.3) \quad \text{Find } p \in H_0^1(\Omega) \quad \text{such that} \quad \int_D \nabla p \cdot \nabla v \, dx = - \int_D v \, dx \quad \text{for all } v \in H_0^1(\Omega).$$

Notice that (3.2) is the so-called volume or weak formulation of the shape derivative of (2.1). Besides the volume formulation, there exists an alternative representation of (3.2) by virtue of the well known Hadamard structure theorem; see [Delfour, Zolésio, 2011](#), Chapter 9, Theorem 3.6. We state it here in a particularized version for problem (2.1). From now on,  $\nu$  denotes the outer unit normal vector along the boundary  $\partial\Omega$  of  $\Omega$ .

**COROLLARY 3.2** (Hadamard structure theorem for (2.1)). *The shape derivative (3.2) of problem (2.1) has the representation*

$$(3.4) \quad J'(\Omega; V) = \int_{\partial\Omega} g_{\Omega}(V \cdot \nu) \, ds \quad \text{with } g_{\Omega} = -\frac{\partial u}{\partial \nu} \frac{\partial p}{\partial \nu}.$$

Notice that under the assumption that  $\Omega$  has a  $C^{1,1}$ -boundary,  $u$  and  $p$  belong to  $H^2(\Omega)$  and thus their normal derivatives are in  $H^{1/2}(\partial\Omega)$ , which embeds into  $L^4(\partial\Omega)$  when  $d \leq 3$ ; see for instance [Adams, Fournier, 2003](#), Theorem 4.12. Consequently,  $g_{\Omega} = -\frac{\partial u}{\partial \nu} \frac{\partial p}{\partial \nu}$  belongs to  $L^2(\partial\Omega)$  in this case.

Formula (3.4) is known as the boundary or strong representation of (3.2), and it can be obtained from (3.2) by the divergence theorem; compare [Sturm, 2015](#), [Sokołowski, Zolésio, 1992](#), Chapter 3.3, [Haslinger, Mäkinen, 2003](#), Example 3.3. We also refer the reader to [Hiptmair, Paganini, Sargheini, 2015](#), where the volume and boundary formulations are compared w.r.t. their order of convergence in a finite element setting.

**4. Investigation of the Discrete Objective.** In order to solve the shape optimization problem (2.1) numerically, some kind of discretization has to be applied. The most common choice in the literature consists in a discretization of the PDE by some finite element space defined over a computational mesh, which we denote by  $\Omega_h$  and whose nodal positions serve to represent the discrete unknown domain.

A common choice is to replace  $H_0^1(\Omega)$  by the finite element space of piecewise linear, globally continuous functions,

$$(4.1) \quad S_0^1(\Omega_h) = \{u \in H_0^1(\Omega_h) : u|_T \in \mathcal{P}_1(T) \text{ for all cells } T \text{ in } \Omega_h\}$$



defined over an approximation  $\Omega_h$  of  $\Omega$  consisting of geometrically conforming simplicial cells, i.e., triangles and tetrahedra in  $d = 2$  or  $d = 3$  space dimensions, respectively. Consequently, the state equation (2.2) is replaced by

$$(4.2) \quad \text{Find } u_h \in S_0^1(\Omega_h) \quad \text{such that} \quad \int_{\Omega} \nabla u_h \cdot \nabla v_h \, dx = \int_{\Omega} f v_h \, dx \quad \forall v_h \in S_0^1(\Omega_h).$$

This leads to the following discrete version of (2.1) frequently encountered in the literature,

$$(4.3) \quad \begin{aligned} &\text{Minimize} \quad \int_{\Omega_h} u_h \, dx \quad \text{w.r.t. } u_h \in S_0^1(\Omega_h) \text{ and the nodal positions in } \Omega_h \\ &\text{s.t.} \quad (4.2). \end{aligned}$$

We refer the reader to [Gangl et al., 2015](#); [Sturm, 2016](#); [Schulz, Siebenborn, Welker, 2016](#); [Schulz, Siebenborn, 2016](#) for examples of this procedure.

Let us denote by  $J_h(\Omega_h)$  the reduced objective value in (4.3), i.e.,  $J_h(\Omega_h) = \int_{\Omega_h} u_h \, dx$ , where  $u_h$  is the unique solution of (4.2). In order to derive a discrete variant of the volume formulation (3.2) of the shape derivative, we introduce the discrete adjoint equation,

$$(4.4) \quad \text{Find } p_h \in S_0^1(\Omega_h) \quad \text{such that} \quad \int_{\Omega_h} \nabla p_h \cdot \nabla v_h \, dx = - \int_{\Omega_h} v_h \, dx \quad \text{for all } v_h \in S_0^1(\Omega_h).$$

The following theorem shows that a straightforward replacement of the state  $u$  and adjoint state  $p$  by their finite element equivalents  $u_h$  and  $p_h$  in (3.2) yields the correct formula for the shape derivative  $J'_h(\Omega_h; V_h)$  of the discrete objective  $J_h$ , provided that the perturbation field  $V_h$  is piecewise linear, i.e.,  $V_h$  belongs to

$$(4.5) \quad S^1(\Omega_h)^d = \{u \in H^1(\Omega_h)^d : u|_T \in \mathcal{P}_1(T)^d \text{ for all cells } T \text{ in } \Omega_h\}.$$

**THEOREM 4.1.** *Suppose that  $u_h$  and  $p_h$  are the unique weak solutions of the discrete state equation (4.2), and the discrete adjoint equation (4.4), respectively. Moreover, let  $V_h \in S^1(\Omega_h)^d$ . Then*

$$(4.6) \quad \begin{aligned} J'_h(\Omega_h; V_h) &= \int_{\Omega_h} u_h (\operatorname{div} V_h) \, dx \\ &\quad + \int_{\Omega_h} (\nabla u_h)^\top [(\operatorname{div} V_h) \operatorname{id} - DV_h - DV_h^\top] \nabla p_h \, dx - \int_{\Omega_h} \operatorname{div}(f V_h) p_h \, dx. \end{aligned}$$

The proof of this theorem follows along the lines of the continuous case, see, e.g., [Hiptmair, Paganini, Sargheini, 2015](#); [Laurain, Sturm, 2016](#). A detailed derivation can be found in [Delfour, Payre, Zolesio, 1985](#), Section 4.

- Remark 4.2.*
1. [Theorem 4.1](#) can be viewed as the statement that discretization and optimization (in the sense of forming the shape derivative) commute for problem (2.1).
  2. The finite element analogue of the *boundary expression* (3.4) is *not* an exact representation of the discrete shape derivative. This is since the integration by parts necessary to pass from the volume to the boundary expression has to be done element by element and it leaves inter-element contributions; see also the discussion in [Berggren, 2010](#).



3. [Theorem 4.1](#) remains true when higher order Lagrangian finite elements on simplices are used in place of  $S_0^1(\Omega_h)$ . However it is essential that  $V_h$  remains piecewise linear so piecewise polynomials are transformed into piecewise polynomials of the same order.
4. Alternative expressions for (4.6) can be obtained following the so-called discrete adjoint approach, in which the derivative of  $J_h(\Omega_h)$  w.r.t. the nodal positions of  $\Omega_h$  is addressed by differentiating the finite element matrices. We refer to [Schneider, Jimack, 2008](#); [Berggren, 2010](#); [Roth, Ulbrich, 2013](#) for examples of this procedure.

Despite the simplicity to obtain the shape derivative of the discrete problem, we would like to emphasize here that the discrete problem (4.3) itself has the following serious drawback. The search space obtained from utilizing the nodal positions of the mesh  $\Omega_h$  as optimization variables includes meshes with very degenerate cells. Those lead to poor approximations of solutions of the state equation, which may give rise, however, to smaller values of the discrete objective. Therefore, any optimization algorithm for the solution of (4.3) sooner or later is likely to encounter spurious descent directions which typically have support in only a few mesh nodes and which lead to degenerate meshes.

**EXAMPLE 4.3.** *Let us illustrate this behavior by means of problem (2.1) with data  $f(x, y) = 2.5(x + 0.4 - y^2)^2 + x^2 + y^2 - 1$ . The optimal domain  $\Omega$  is unknown. We begin with the computational mesh  $\Omega_h$  shown in [Figure 2](#) (left). Consider for example the piecewise linear vector field  $V_h$  represented by its nodal values*

$$V_h = \begin{cases} (-0.9510, -0.3090)^\top, & \text{for the node } v_0, \\ (0, 0)^\top, & \text{for all other nodes} \end{cases}$$

where the boundary node  $v_0$  can be easily identified from [Figure 2](#).

We found that  $V$  is not only a descent direction for the objective at  $\Omega_h$  but in fact that the line search function

$$\alpha \mapsto J(T_\alpha(\Omega_h)), \quad T_\alpha = \text{id} + \alpha V_h$$

decreases until the triangle formed by  $v_0$  and its two interior neighbors degenerates to a line, which happens at  $\alpha = 0.1$ ; see [Figure 1](#). At this point, finite element computations break down.

In computational experience spurious descent directions do not usually occur during the early iterates. Thus they can be, and often are, avoided by early stopping, at the expense of a reduced tolerance. Alternatively, mesh quality control and remeshing can help to avoid mesh destruction, but this introduces discontinuities in the objective function's history.

In any case, the existence of spurious descent directions is a structural disadvantage of problem (4.3). Therefore we propose in the following section a new computational approach. Our approach does not seek to solve (4.3) literally but in a certain relaxed sense, which is inspired by the Hadamard structure theorem and which avoids spurious descent directions.

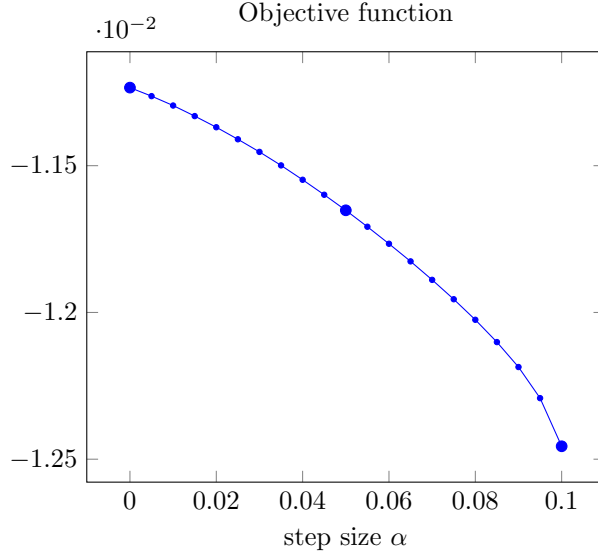


Fig. 1:  $\alpha \mapsto J(T_\alpha(\Omega_h))$  for [Example 4.3](#). The step sizes  $\alpha = 0.00$ ,  $\alpha = 0.05$ ,  $\alpha = 0.10$  belonging to the domains in [Figure 2](#) are highlighted.

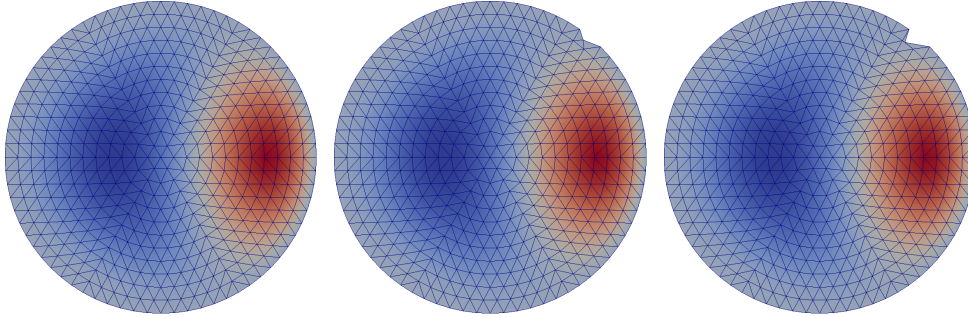


Fig. 2: Evolution of the mesh under  $\alpha \mapsto T_\alpha(\Omega_h)$  with perturbation field  $V_h$  given in [Example 4.3](#) at  $\alpha = 0.00$ ,  $\alpha = 0.05$ ,  $\alpha = 0.10$  (from left to right). The solutions  $u_h$  of the state equation (4.2) are also shown.

**5. Restricted Mesh Deformations.** By the Hadamard structure theorem, the shape derivative for the continuous problem consists of normal boundary forces only, see (3.4) above. This is no longer the case for the discrete problem. The reason is that the finite element solutions  $u_h$  and  $p_h$  are only of limited regularity, and thus a global integration by parts necessary to pass from the volume expression (4.6) to a boundary expression is not available. This has been pointed out, for instance, in [Delfour, Zolésio, 2011](#), note on p. 562. Therefore, we are going to continue with the discretely exact volume expression (4.6) but mimic the behavior of the continuous setting in the evaluation of the shape gradient, where we allow only for shape displacements which are induced by normal forces.

**5.1. Continuous Setting.** To illustrate the situation, we start by discussing the continuous case. We have seen in (3.2) that the shape derivative  $J'(\Omega; \cdot)$  is an element of a dual space, e.g. an element of  $(W^{1,\infty}(\Omega)^d)^*$ . In order to utilize this information for moving the domain  $\Omega$ , we have to convert this dual element into a proper function. We follow the approach of [Schulz, Siebenborn, Welker, 2016](#). To this end, we introduce the elasticity operator  $E : H^1(\Omega)^d \rightarrow (H^1(\Omega)^d)^*$  via

$$(5.1) \quad \langle EV, W \rangle := \int_{\Omega} 2\mu \varepsilon(V) : \varepsilon(W) + \lambda \operatorname{trace}(\varepsilon(V)) \operatorname{trace}(\varepsilon(W)) + \delta V \cdot W \, dx$$

for all  $V, W \in H^1(\Omega)^d$ . Here and throughout,  $D$  denotes the derivative (Jacobian) of a vector valued function,  $\varepsilon(V) = (DV + DV^\top)/2$  is the linearized strain tensor,  $\mu, \lambda$  are the Lamé parameters and  $\delta > 0$  is a damping term. We assume  $\mu > 0$ ,  $d\lambda + 2\mu > 0$  so that  $E$  becomes positive semi-definite on  $H^1(\Omega)^d$ . Note that we do not consider Dirichlet boundary conditions in the space  $H^1(\Omega)^d$ . Therefore a positive damping parameter  $\delta > 0$  is needed to ensure the coercivity of  $E$ , i.e.,  $\langle EV, V \rangle \geq \underline{c} \|V\|_{H^1(\Omega)^d}^2$  with some  $\underline{c} > 0$ . This result is due to Korn's inequality, see for instance [Attouch, Buttazzo, Michaille, 2014](#), Proposition 6.6.1. Thus,  $E$  is an isomorphism and it furnishes  $H^1(\Omega)^d$  with an inner product  $(V, W)_E := \langle EV, W \rangle$  so that  $E$  becomes the associated Riesz isomorphism.

In order to avoid technical regularity issues, we assume that the shape derivative (3.2) enjoys the higher regularity  $J'(\Omega; \cdot) \in (H^1(\Omega)^d)^*$ . This holds, e.g., if  $\Omega$  is sufficiently smooth, due to the higher regularity of  $u$  and  $p$ . In order to compute the negative shape gradient w.r.t. the  $E$ -inner product on the continuous level, we solve

$$(5.2) \quad \text{Minimize} \quad J'(\Omega; V) + \frac{1}{2} \langle EV, V \rangle \quad \text{s.t. } V \in H^1(\Omega)^d.$$

The solution of this problem yields the negative shape gradient

$$(5.3) \quad V_{\text{grad}} := -E^{-1} J'(\Omega; \cdot).$$

Now, we introduce the normal force operator  $N : L^2(\partial\Omega) \rightarrow (H^1(\Omega)^d)^*$  given by

$$(5.4) \quad \langle NF, V \rangle = \int_{\partial\Omega} F(V \cdot \nu) \, ds$$

for all  $F \in L^2(\partial\Omega)$  and  $V \in H^1(\Omega)^d$ . Using again (3.4), we find that  $J'(\Omega; \cdot)$  can be written as  $J'(\Omega; \cdot) = Ng_\Omega$  with

$$g_\Omega = -\frac{\partial u}{\partial \nu} \frac{\partial p}{\partial \nu} \in L^2(\partial\Omega).$$

Therefore, it is easy to see that problem (5.2) is equivalent to

$$(5.5) \quad \begin{aligned} &\text{Minimize} \quad J'(\Omega; V) + \frac{1}{2} \langle EV, V \rangle \\ &\text{with respect to} \quad V \in H^1(\Omega)^d, F \in L^2(\partial\Omega) \\ &\text{such that} \quad EV - NF = 0. \end{aligned}$$

Indeed, the additional constraint  $EV - NF = 0$  is automatically satisfied by the unconstrained solution of (5.2). However, we will see that this property is lost

after discretization, i.e., the discrete counterparts of (5.2) and (5.5) are going to differ. Note that the solution  $(V, F)$  of (5.5) is unique due to coercivity of  $E$  and injectivity of  $N$ . Moreover, since  $\begin{bmatrix} E & -N \end{bmatrix}$  is surjective, there exists a unique Lagrange multiplier  $\Pi \in H^1(\Omega)^d$  associated with the constraint  $EV - NF = 0$ ; see for instance [Luenberger, 1969](#), Chapter 9.3, Theorem 1. We therefore obtain the following necessary and sufficient optimality conditions for (5.5) in saddle-point form,

$$(5.6) \quad \begin{pmatrix} E & 0 & E \\ 0 & 0 & -N^* \\ E & -N & 0 \end{pmatrix} \begin{pmatrix} V \\ F \\ \Pi \end{pmatrix} = \begin{pmatrix} -J'(\Omega; \cdot) \\ 0 \\ 0 \end{pmatrix}.$$

Here,  $N^* : H^1(\Omega)^d \rightarrow L^2(\partial\Omega)$  is the adjoint of  $N$ , where we identified  $L^2(\partial\Omega)$  with its dual space. The multiplier  $\Pi$  in (5.6) necessarily satisfies  $\Pi = 0$  since  $E$  is bijective. Now, it is easy to see that (5.6) is equivalent to solving

$$(5.7a) \quad \begin{pmatrix} 0 & N^* \\ N & E \end{pmatrix} \begin{pmatrix} F \\ \Pi \end{pmatrix} = \begin{pmatrix} 0 \\ -J'(\Omega; \cdot) \end{pmatrix},$$

$$(5.7b) \quad V = -E^{-1}J'(\Omega; \cdot) - \Pi.$$

Recall that  $-E^{-1}J'(\Omega; \cdot)$  is the usual negative shape gradient w.r.t.  $E$  (i.e., the solution of (5.2)), whereas  $-\Pi$  is a correction in order to obtain a shape displacement in the subspace  $\text{im}(E^{-1}N)$ . Again, we emphasize that we have  $\Pi = 0$  in the continuous setting, due to  $-J'(\Omega; \cdot) = -N g_\Omega$ . Therefore, the solution of (5.7) is just the usual shape gradient  $V_{\text{grad}} = -E^{-1}J'(\Omega; \cdot)$ .

Before discussing the discretized setting, we note that (5.5) is equivalent to

$$(5.8) \quad \begin{aligned} & \text{Minimize} \quad \frac{1}{2} \langle E(V - V_{\text{grad}}), V - V_{\text{grad}} \rangle \\ & \text{with respect to} \quad V \in H^1(\Omega)^d, F \in L^2(\partial\Omega) \\ & \text{such that} \quad EV - NF = 0. \end{aligned}$$

Hence, the solution  $V$  is the orthogonal projection (w.r.t. the inner product induced by  $E$ ) of the usual shape gradient  $V_{\text{grad}} = -E^{-1}J'(\Omega; \cdot)$  into the space  $\text{im}(E^{-1}N)$ , i.e., the space of deformations induced by normal forces. This motivates to denote the solution of (5.5) by  $V_{\text{proj grad}}$ .

**5.2. Discretized Setting.** Next, we discuss the discretized setting. We refer to [section 4](#) above for the introduction of the finite-element discretization. In addition to the FE space  $S_0^1(\Omega_h) \subset H_0^1(\Omega_h)$ , we recall from (4.5) the discrete space of mesh deformations

$$S^1(\Omega_h)^d = \{u \in H^1(\Omega_h)^d : u|_T \in \mathcal{P}_1(T)^d \text{ for all cells } T \text{ in } \Omega_h\}$$

and the boundary space

$$(5.9) \quad S^1(\partial\Omega_h) = \{u \in C(\partial\Omega_h)^d : u|_E \in \mathcal{P}_1(E)^d \text{ for all edges } E \text{ on } \partial\Omega_h\}.$$

We recall that the discrete shape derivative  $J'_h(\Omega_h; \cdot) \in (S^1(\Omega_h)^d)^*$  was given in (4.6). Moreover, the discretization directly leads to the discretized operators  $E_h : S^1(\Omega_h)^d \rightarrow$

$(S^1(\Omega_h)^d)^*$ ,  $N_h : S^1(\partial\Omega_h) \rightarrow (S^1(\Omega_h)^d)^*$  which are defined via

$$\begin{aligned}\langle E_h V_h, W_h \rangle &:= \int_{\Omega_h} 2\mu \varepsilon(V_h) : \varepsilon(W_h) + \lambda \operatorname{trace}(\varepsilon(V_h)) \operatorname{trace}(\varepsilon(W_h)) + \delta V_h \cdot W_h \, dx, \\ \langle N_h F_h, V_h \rangle &:= \int_{\partial\Omega_h} F_h (V_h \cdot \nu) \, ds\end{aligned}$$

for all  $V_h, W_h \in S^1(\Omega_h)^d$  and  $F_h \in S^1(\partial\Omega_h)$ . Next, we will investigate the discrete counterparts of (5.2) and (5.5). The straightforward discretization of (5.2) reads

$$(5.10) \quad \text{Minimize} \quad J'_h(\Omega_h; V_h) + \frac{1}{2} \langle E_h V_h, V_h \rangle.$$

We denote its unique solution by  $V_{\text{grad},h}$ .

The important difference to the continuous case is that Hadamard's structure theorem is not available. The reason is that the discrete state  $u_h$  has only the limited regularity  $u_h \in H_0^1(\Omega_h)$  and this regularity is not enough to transform the domain integral into a boundary integral via integration by parts, see the last paragraph in chapter 10, section 5.6 of [Delfour, Zolésio, 2011](#). Therefore, unlike in the continuous case,  $J'_h(\Omega_h; \cdot)$  does not belong, in general, to the image space of  $N_h$ . Consequently, the solution  $V_h$  of (5.10) has contributions not only from normal forces in the shape derivative  $J'_h(\Omega_h; \cdot)$ , but also from interior forces as well as tangential boundary forces. Numerical examples in [section 6](#) will show that these interior and tangential forces are responsible for spurious descent directions, which in turn lead to degenerate meshes.

Therefore, we conclude that it is not reasonable to try to solve

$$\text{Minimize } J_h(\Omega_h)$$

or its stationarity condition

$$(5.11) \quad \text{Find a triangulation } \Omega_h \text{ such that } V_{\text{grad},h} = -E_h^{-1} J'_h(\Omega_h; \cdot) = 0$$

as a discretization of the continuous problem (1.1).

Hence, we consider the discretization of (5.5)

$$(5.12) \quad \begin{aligned} &\text{Minimize} \quad J'_h(\Omega_h; V_h) + \frac{1}{2} \langle E_h V_h, V_h \rangle \\ &\text{with respect to} \quad V_h \in S^1(\Omega_h)^d, F_h \in S^1(\partial\Omega_h) \\ &\text{such that} \quad E_h V_h - N_h F_h = 0. \end{aligned}$$

in which we restrict  $E_h V_h$  to the image space of the discrete normal force operator  $N_h$ . As in the continuous setting, this problem is equivalent to the solution of

$$(5.13) \quad \begin{pmatrix} E_h & 0 & E_h \\ 0 & 0 & -N_h^* \\ E_h & -N_h & 0 \end{pmatrix} \begin{pmatrix} V_h \\ F_h \\ \Pi_h \end{pmatrix} = \begin{pmatrix} -J'_h(\Omega_h; \cdot) \\ 0 \\ 0 \end{pmatrix}.$$

It is clear that (5.13) can also be reduced as in (5.7). For later reference, we mention that the solution  $(V_{\text{proj grad},h}, F_h, \Pi_h)$  of (5.13) satisfies

$$(5.14) \quad \begin{aligned} \langle E_h V_{\text{proj grad},h}, V_{\text{proj grad},h} \rangle &= -\langle E_h V_{\text{proj grad},h}, \Pi_h \rangle - J'_h(\Omega_h; V_{\text{proj grad},h}) \\ &= -\langle N_h F_h, \Pi_h \rangle - J'_h(\Omega_h; V_{\text{proj grad},h}) \\ &= -J'_h(\Omega_h; V_{\text{proj grad},h}) \end{aligned}$$

since  $N_h^* \Pi_h = 0$  holds. This shows that  $V_{\text{proj grad},h}$  is always a descent direction for the discrete objective  $J_h(\Omega_h; \cdot)$ .

As we have seen in (5.8) for the continuous setting, the solution  $V_h$  of (5.12) also solves

$$(5.15) \quad \begin{aligned} & \text{Minimize} && \frac{1}{2} \langle E_h(V_h - V_{\text{grad},h}), V_h - V_{\text{grad},h} \rangle \\ & \text{with respect to} && V_h \in S^1(\Omega_h)^d, F_h \in S^1(\partial\Omega_h) \\ & \text{such that} && E_h V_h - N_h F_h = 0, \end{aligned}$$

where  $V_{\text{grad},h} = -E_h^{-1} J'_h(\Omega_h; \cdot)$  is the solution of (5.10). Again, the solution  $V_{\text{proj grad},h}$  of (5.15) can be interpreted as the projection (w.r.t. the  $E_h$  inner product) of  $V_{\text{grad},h}$  onto the image space of  $E_h^{-1} N_h$ . Therefore, the notation  $V_{\text{proj grad},h}$  for the solution of (5.12) is justified.

Our main idea is now to propose, instead of (5.11),

$$(5.16) \quad \text{Find a triangulation } \Omega_h \text{ such that } V_{\text{proj grad},h} = 0$$

as an appropriate discrete version of (1.1). Note that this is fundamentally different from the ad-hoc discretization (5.11) since we neglect the contributions of  $J'_h(\Omega_h; \cdot)$  which do not belong to the image space of  $N_h$ . We will see via numerical examples that this problem (5.16) can be solved to high accuracy by an iterative algorithm using the solution  $V_{\text{proj grad},h}$  of (5.12) for the displacement of the triangulation  $\Omega_h$  (together with a line search).

For later use, we are going to characterize stationarity of  $\Omega_h$  in the sense of (5.16). The deformation  $V_h = 0$  solves the projection problem (5.15) if and only if

$$\langle E_h V_{\text{grad},h}, E_h^{-1} N_h F_h \rangle = 0 \quad \forall F_h \in S^1(\partial\Omega_h).$$

This, in turn, is equivalent to

$$(5.17) \quad \int_{\partial\Omega_h} F_h(V_{\text{grad},h} \cdot \nu) \, ds = 0 \quad \forall F_h \in S^1(\partial\Omega_h).$$

This means that  $\Omega_h$  is stationary in the sense of (5.16) if and only if the usual shape gradient  $V_{\text{grad},h}$  is a tangential vector field on  $\Omega_h$  in a discrete sense.

We can now state a restricted gradient algorithm for the solution of (5.16), where we use  $V_{\text{proj grad},h}$  as the deformation field which provides the search direction in the domain transformation. It is sufficient to utilize a simple backtracking strategy to comply with the Armijo condition

$$(5.18) \quad J_h((\text{id} + \alpha V_{\text{proj grad},h})(\Omega_h)) \leq J_h(\Omega_h) + \sigma \alpha J'_h(\Omega_h; V_{\text{proj grad},h}).$$

Here,  $\sigma \in (0, 1)$  is a parameter.

Since we are using the perturbation of identity approach (1.2) instead of a more sophisticated family of domain transformations, we also perform a mesh quality control in order to avoid gradient steps which are too large. To this end, we check that the conditions

$$(5.19) \quad \frac{1}{2} \leq \det(\text{id} + \alpha DV_{\text{proj grad},h}) \leq 2, \quad \|\alpha DV_{\text{proj grad},h}\|_F \leq 0.3$$

are satisfied in every cell throughout the entire domain. Here,  $\|\cdot\|_F$  denotes the Frobenius norm of matrices. The first condition monitors the change of volume of the cell, while the second additionally inhibits large changes of the angles. Note that this amounts to checking three inequalities per cell. Due to (5.14), we use

$$(5.20) \quad \langle E_h V_{\text{proj grad},h}, V_{\text{proj grad},h} \rangle = -J'_h(\Omega_h; V_{\text{proj grad},h}) \leq \varepsilon_{\text{tol}}^2$$

as a convergence criterion for some small  $\varepsilon_{\text{tol}} > 0$ . These considerations lead to [Algorithm 1](#).

---

**Algorithm 1:** Restricted gradient method for (5.16).

---

**Data:** Initial domain  $\Omega_h$   
Initial step size  $\alpha$ , convergence tolerance  $\varepsilon_{\text{tol}}$ ,  
line search parameters  $\beta \in (0, 1)$ ,  $\sigma \in (0, 1)$   
**Result:** Improved domain  $\Omega_h$  on which (5.16) holds up to  $\varepsilon_{\text{tol}}$

```

1 for  $i \leftarrow 1$  to  $\infty$  do
2   Solve the discrete state equation (4.2) for  $u_h$ ;
3   Solve the discrete adjoint equation (4.4) for  $p_h$ ;
4   Solve (5.12) for  $V_{\text{proj grad},h}$  with shape derivative  $J'(\Omega_h; \cdot)$  from (4.6);
5   if  $\langle E_h V_{\text{proj grad},h}, V_{\text{proj grad},h} \rangle \leq \varepsilon_{\text{tol}}^2$  then
6     STOP, the current iterate  $\Omega_h$  is almost stationary for (5.16);
7   end
8   Increase step size  $\alpha \leftarrow \alpha/\beta$ ;
9   while (5.18) or (5.19) is violated do
10    Decrease step size  $\alpha \leftarrow \beta \alpha$ ;
11  end
12  Transform the domain according to  $\Omega_h \leftarrow (\text{id} + \alpha V_{\text{proj grad},h})(\Omega_h)$ ;
13 end

```

---

**5.3. Comparison with the approach of Schulz, Siebenborn, Welker, 2016.** In Schulz, Siebenborn, Welker, 2016, the authors propose a different way to convert  $J'_h(\Omega_h; \cdot)$  into a deformation field from  $S^1(\Omega_h)^d$ . Instead of solving (5.10) directly, i.e.,

$$\text{Find } V_{\text{grad},h} \in S^1(\Omega_h)^d \quad \text{s.t.} \quad E_h V_{\text{grad},h} = -J'_h(\Omega_h, W_h) \quad \forall W_h \in S^1(\Omega_h)^d,$$

they propose that “Only test functions whose support includes  $\Gamma_{\text{int}}$  are considered [...]”, see Schulz, Siebenborn, Welker, 2016, p. 2813. In their problem formulation,  $\Gamma_{\text{int}}$  corresponds to  $\partial\Omega_h$  in our formulation. We interpret this as follows. We denote by  $D_h : S^1(\Omega_h)^d \rightarrow S_0^1(\Omega_h)^d$  the projection operator defined via

$$(D_h W_h)(x) = \begin{cases} W_h(x) & \text{if } x \text{ is an interior node of } \Omega_h \\ 0 & \text{if } x \text{ is a boundary node of } \Omega_h \end{cases}$$

for all nodes  $x$  from the mesh  $\Omega_h$ . Note that  $D_h$  can be represented by a diagonal matrix (with entries 0 and 1) in the standard basis of  $S^1(\Omega_h)^d$ . Then, the deformation  $V_{h,\text{SSW}}$  is computed via the solution of

$$\text{Find } V_{h,\text{SSW}} \in S^1(\Omega_h)^d \quad \text{s.t.} \quad E_h V_{h,\text{SSW}} = -J'_h(\Omega_h, D_h W_h) \quad \forall W_h \in S^1(\Omega_h)^d.$$

We compare this suggestion with our approach.



- The deformation  $V_{h,\text{SSW}}$  can be computed faster than  $V_{\text{proj grad},h}$ , since the linear system is smaller than (5.13).
- The deformation  $V_{h,\text{SSW}}$  cannot be understood as a (negative) gradient direction generated by an inner product on (a subspace of)  $S^1(\Omega_h)^d$ . What is more,  $V_{h,\text{SSW}}$  may fail to be a descent direction, since

$$J'_h(\Omega_h, V_{h,\text{SSW}}) = -J'_h(\Omega_h, E_h^{-1} J'_h(\Omega_h, D_h \cdot))$$

might be positive. This indeed does happen in our numerical examples, see [section 6](#) below. In contrast, our suggestion  $V_{\text{proj grad},h}$  is generated by an inner product on the subspace  $\{V_h \in S^1(\Omega_h)^d : \exists F_h \in S^1(\partial\Omega_h) : E_h V_h = N_h F_h\}$ .

- The deformation  $V_{h,\text{SSW}}$  is induced by the forces corresponding to the linear map  $W_h \mapsto -J'_h(\Omega_h, D_h W_h)$ . Due to the operator  $D_h$ , these forces only act on the boundary  $\partial\Omega_h$ . However, these forces may contain tangential components and this is a crucial difference to our approach. Eventually, this will generate tangential movement of boundary points leading to a deterioration of the mesh quality. We see the onset of this in [Figure 3](#) (center bottom), where boundary nodes accumulate on the right part of the boundary and a rarefaction of nodes occurs on the left.

**6. Numerical Results: Comparison of Gradient Methods.** The main goal of this section is to compare our proposed restricted gradient method, see [Algorithm 1](#), to a classical shape gradient method. The latter is identical to [Algorithm 1](#) except that  $V_{\text{proj grad},h}$  is replaced everywhere by the negative shape gradient  $V_{\text{grad},h}$  from (5.10). In addition, we also compare it to a method utilizing the deformation fields  $V_{h,\text{SSW}}$  obtained along the lines of [Schulz, Siebenborn, Welker, 2016](#). We refer to the latter as a gradient-like method since it may fail to produce descent directions. Consequently,  $V_{h,\text{SSW}}$  cannot be a negative gradient direction w.r.t. any inner product in this situation.

We consider our model problem (2.1) with data  $f$  as in [Example 4.3](#). The line search parameters  $\beta = 0.5$  and  $\sigma = 0.1$  are used and the initial step size is chosen as  $\alpha = 1$ . For the Lamé and damping parameters in the elasticity operator (5.1) we choose

$$\mu = \frac{E_0}{2(1+\nu)}, \quad \lambda = \frac{E_0 \nu}{(1+\nu)(1-2\nu)}, \quad \delta = 0.2 E_0$$

where  $E_0 = 1.0$  is Young's modulus and  $\nu = 0.4$  is the Poisson ratio. The initial shape for all three methods is the same as in [Figure 2](#) (left). For this first result, the mesh has 864 triangles and 469 vertices but computations on refined meshes are reported below in [Table 1](#).

We implemented the restricted gradient method, [Algorithm 1](#), its classical counterpart as well as the gradient-like method from [Schulz, Siebenborn, Welker, 2016](#) in FENICS, version 2018.1 ([Logg, Mardal, Wells, et al., 2012](#)). We report computational results obtained on a machine with an Intel(R) Xeon(R) CPU E5-4640 at 2.4 GHz.

Our implementation is freely available on GitHub, see [Etling et al., 2018](#). All derivatives were automatically generated by the built-in algorithmic differentiation capabilities of FENICS. The restricted shape gradient  $V_{\text{proj grad},h}$ , i.e., the solution of (5.12), was computed via the discrete counterpart of (5.7). The linear system was

solved using SciPy's `spsolve` with the SUPERLU solver (Li, 2005), i.e., with the setting `use_umfpack = False`.

The restricted gradient method reached the desired tolerance

$$(6.1) \quad \|V_{\text{proj grad},h}\|_{E_h} = \sqrt{|J'_h(\Omega_h; V_{\text{proj grad},h})|} \leq \varepsilon_{\text{tol}} = 10^{-7}$$

at iteration 864 after 40 seconds, while the classical gradient method was stopped at iteration 1500 after 43 seconds, where it had only reached

$$\|V_{\text{grad},h}\|_{E_h} = \sqrt{|J'_h(\Omega_h; V_{\text{grad},h})|} \approx 4 \cdot 10^{-3}.$$

On the other hand, the gradient-like method from Schulz, Siebenborn, Welker, 2016 was stopped at iteration 1435 since it failed to generate a descent direction. This process took 49 seconds, and at that point the method had reached

$$\sqrt{|J'_h(\Omega_h; V_{h,\text{SSW}})|} \approx 3 \cdot 10^{-5}.$$

Figure 4 shows the complete history of the objective and respective shape gradient norms. The geometry condition (5.19) was violated only once for all methods, namely in the first very iteration, leading to a reduction of the initial step size. The Armijo condition (5.18) failed approximately once per iteration on average. Typical accepted step sizes were  $\alpha = 0.5$  and occasionally  $\alpha = 1$ .

Figure 3 shows the domains  $\Omega_h$  during the iteration of all three methods for comparison. It can clearly be inferred that the initial iterates are virtually identical. The classical gradient method begins to produce visibly different shapes around iteration 500, when the objective value (shown in Figure 4) has practically converged but the gradient norms are still

$$\|V_{\text{grad},h}\|_{E_h} \approx 5 \cdot 10^{-3} \quad \text{and} \quad \|V_{\text{proj grad},h}\|_{E_h} \approx 4 \cdot 10^{-6},$$

respectively. At this point, the classical gradient method starts to pursue spurious descent directions, which results in a further decrease of the discrete objective at the expense of increasingly degenerate meshes. Similarly, the gradient method from Schulz, Siebenborn, Welker, 2016 produces some tangential movement on the boundary. This decreases the mesh quality slightly and inhibits further decrease of the norm of the gradient. As an indicator for the mesh quality, we used the “minimum radius ratio”, which is provided by FENICS and is defined as the minimum over all cells of two (the geometric dimension) times the inradius divided by the circumradius of the cell. This value lies between 0 and 1, where 0 corresponds to a degenerate cell and 1 to an equilateral triangle. The results for the mesh quality are shown in Figure 5.

We conclude from this numerical experiment that the restricted gradient method is slightly more expensive per iteration compared to the classical gradient method and the one obtaining its search direction from Schulz, Siebenborn, Welker, 2016. However, the proposed restricted gradient method reduces the norm of the restricted gradient much more effectively than the other two methods. The cell aspect ratios for the restricted gradient method and the one from Schulz, Siebenborn, Welker, 2016 are equally good. However, the latter eventually failed to produce descent directions in

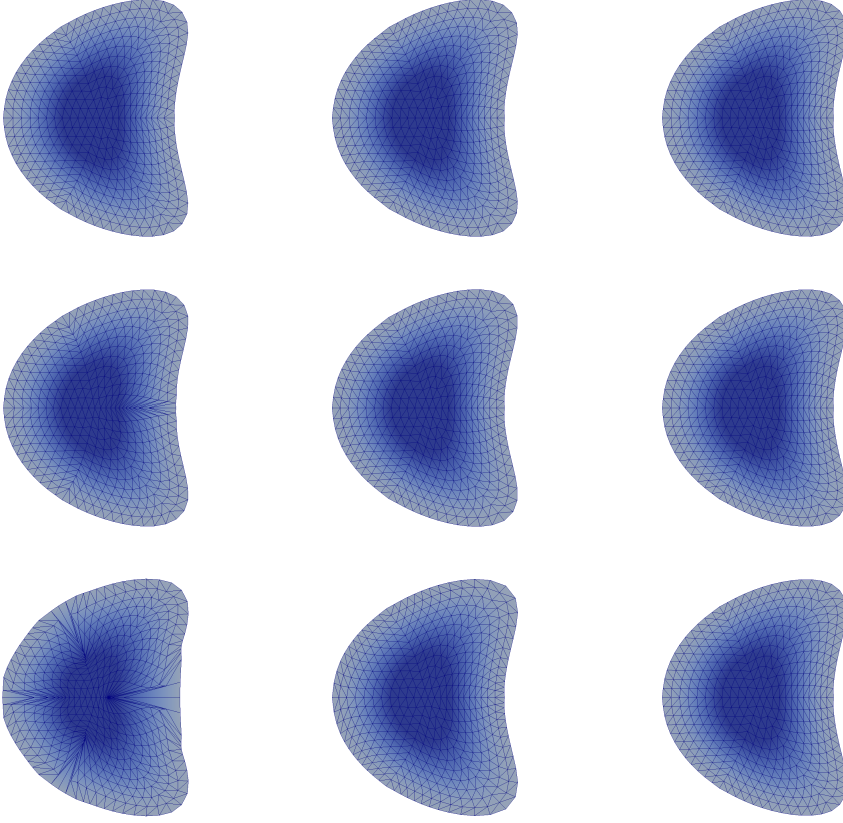


Fig. 3: Intermediate shapes  $\Omega_h$  obtained with the classical gradient method (left), the gradient method from [Schulz, Siebenborn, Welker, 2016](#) (middle) and the restricted gradient method (right) at iterations 300, 600, and 1500, 1435 and 864, respectively.

our experiment, while the classical gradient method created a distorted mesh and did not converge.

To further illustrate this point, we show in [Figure 6](#) visualizations of the shape derivative  $J'_h(\Omega_h; \cdot)$  for the classical and restricted gradient methods; see [\(4.6\)](#). In fact, this is a linear functional on the space of piecewise linear perturbation fields  $V_h \in S^1(\Omega_h)^d$ . In [Figure 6](#) we display the  $S^1(\Omega_h)^d$  representer of  $J'_h(\Omega_h; \cdot)$  w.r.t. the  $L^2$  inner product, i.e., we solve a linear system governed by a block-diagonal mass matrix.

Let us comment on the shape derivative for the restricted gradient method as shown in the right column of [Figure 6](#). It is apparent that the displacement field  $V_{\text{grad},h}$ , i.e., the solution of [\(5.10\)](#), is non-zero and in fact essentially the same for the iterations 300, 600, and 864 shown. However  $V_{\text{grad},h}$  also has essentially no component in the space of deformations induced by normal forces. Therefore its projection into this space, see [\(5.15\)](#), leaves us with a very small norm  $\|V_{\text{proj grad},h}\|_{E_h}$ , as shown in

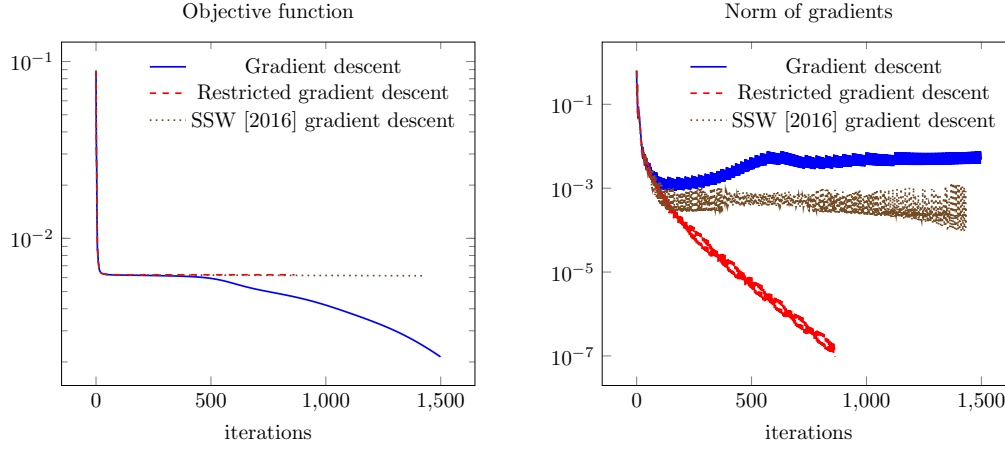


Fig. 4: History of the objective value  $J_h(\Omega_h) + 0.1$  (left) and the norm of the gradients  $\|V_{\text{grad},h}\|_{E_h}$  (for the gradient descent method),  $\|V_{\text{proj grad},h}\|_{E_h}$  (for the restricted gradient method) and  $\sqrt{J'_h(\Omega_h; V_{h,\text{SSW}})}$  (for the gradient-like method from [Schulz, Siebenborn, Welker, 2016](#)) along the iterations (right).

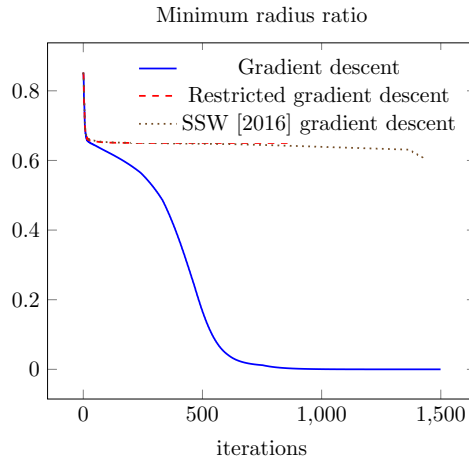


Fig. 5: History of the mesh quality indicator (minimum over all cells of two times the inradius divided by the circumradius of the cell).

[Figure 4](#). The images visualizing the shape derivative for the classical gradient method in the left column of [Figure 6](#) show that the method has allowed the spurious part of the derivative to build up, which eventually dominates the search direction.

**7. Restricted Newton-Like Method.** In the previous two sections we have seen that (5.16) is a reasonable discrete optimality condition and that it can be solved to high accuracy via a first-order gradient descent method. However, as is well known for the minimization of even mildly ill-conditioned quadratic polynomials, gradient descent methods require a large number of iterations to achieve convergence. We

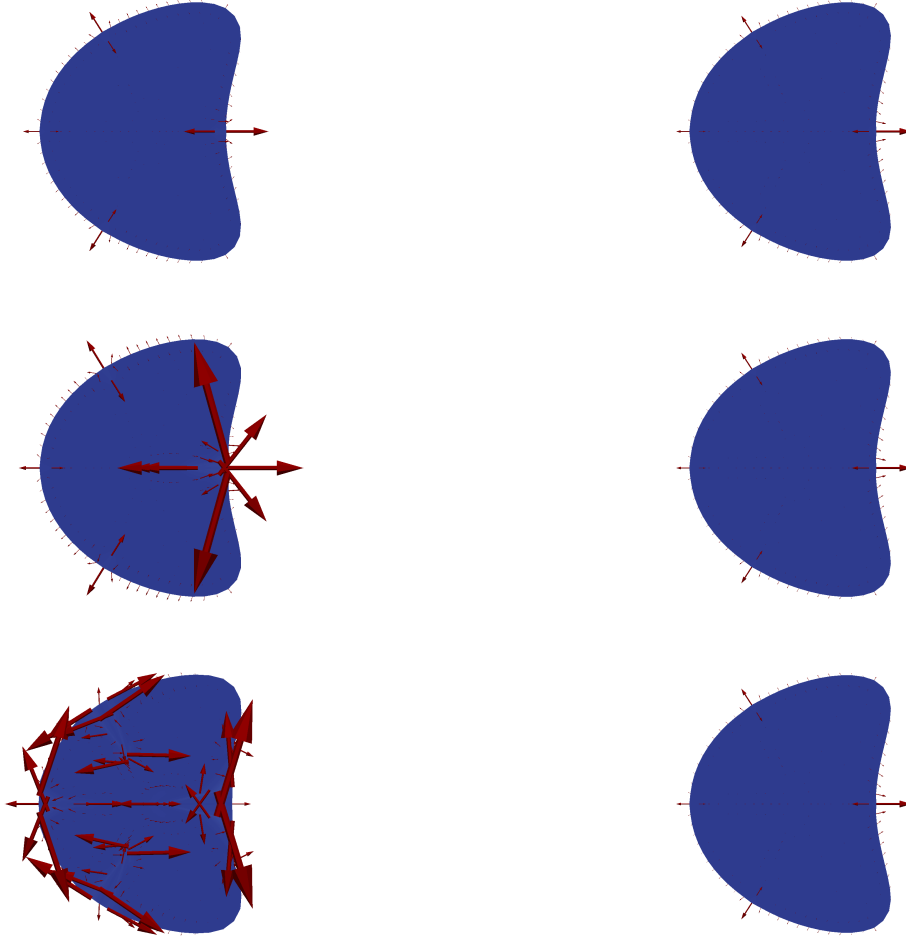


Fig. 6: Visualization of the shape derivatives  $J'_h(\Omega_h; \cdot)$  obtained with the classical gradient method (left), and the restricted gradient method (right) at iterations 300, 600, and 1500, and 864, respectively.

observed the same behavior in [section 6](#).

Therefore, we are also investigating a Newton-like method for solving (5.16). First, we focus on the continuous case and comment on its discretization afterwards. Let  $\Omega$  be our current iterate. As before, we denote by  $u$  the associated state, see (2.2), and by  $p$  the adjoint state, see (3.3). The solution of the restricted shape gradient problem (5.6) at  $\Omega$  is denoted by  $(V_{\text{proj grad}}, F, \Pi)$ . Recall that our goal is to achieve  $V_{\text{proj grad}} = 0$  or, equivalently,  $F = 0$ , cf. (5.16). In practice, we impose a stopping criterion of the form  $\|V_{\text{proj grad}}\|_E \leq \varepsilon_{\text{tol}}$  as we did for the gradient method.

In order to allow the reader to follow the derivation for the solution of (5.16) of our Newton method more easily, we draw the parallel with Newton's method for  $\Phi(x) = 0$  for some  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . We consider the equation  $\Phi(x + \delta x) = 0$  for the

unknown update  $\delta x$ . In our context the iterate  $x$  represents the current domain  $\Omega$  and the update corresponds to a perturbation field  $W$ . Since the update takes  $\Omega$  into a new domain, we need to manipulate the expression  $\Phi(x + \delta x) = 0$  and pull it back to  $\Omega$ . Finally, we linearize about  $\delta x = 0$ , which amounts to  $\Phi(x) + D\Phi(x) \delta x = 0$ .

In our Newton method we seek a deformation field  $W$  (taking the role of  $\delta x$  above) such that the updated domain  $\Omega_W := (\text{id} + W)(\Omega)$  is stationary in the sense that the solution of (5.6) (at  $\Omega_W$  instead of  $\Omega$ ) satisfies  $V_{\text{proj grad}}^W = 0$ . As in section 5 we are only considering updates  $W$  which are induced by a normal force  $G$ , i.e.,  $E W - N G = 0$  should hold.

In order to characterize the stationarity of the transformed domain  $\Omega_W$ , we introduce the elasticity operator  $E_W : H^1(\Omega_W)^d \rightarrow (H^1(\Omega_W)^d)^*$  and the normal force operator  $N_W : L^2(\partial\Omega_W) \rightarrow (H^1(\Omega_W)^d)^*$  on  $\Omega_W$  analogously to (5.1) and (5.4). With the transformation field  $T_W := \text{id} + W : \Omega \rightarrow \Omega_W$ , we define the pull-backs  $E^W : H^1(\Omega)^d \rightarrow (H^1(\Omega)^d)^*$  of  $E_W$  and  $N^W : L^2(\partial\Omega) \rightarrow (H^1(\Omega)^d)^*$  of  $N_W$  via

$$\begin{aligned} \langle E^W W_1, W_2 \rangle &:= \langle E_W (W_1 \circ T_W^{-1}), W_2 \circ T_W^{-1} \rangle, \\ \langle N^W F, W_2 \rangle &:= \langle N_W (F \circ T_W^{-1}), W_2 \circ T_W^{-1} \rangle \end{aligned}$$

for  $W_1, W_2, W \in H^1(\Omega)^d$ .

Since we wish to achieve conditions defined on the current domain  $\Omega$ , rather than on the unknown transformed domain  $\Omega_W$  after the Newton step, we consider the Lagrangian associated with problem (2.1) on  $\Omega_W = T_W(\Omega)$  and pull it back to  $\Omega$ . Using the usual integral substitution and chain rule, and denoting the pulled-back solutions of the state and adjoint equations on  $\Omega_W$  by  $u^W$  and  $p^W$ , respectively, we obtain  $\mathcal{L} : W^{1,\infty}(\Omega)^d \times H_0^1(\Omega) \times H_0^1(\Omega) \rightarrow \mathbb{R}$ , defined as

$$\begin{aligned} \mathcal{L}(W, u^W, p^W) &= \int_{\Omega} u^W \det(\text{id} + DW) dx \\ &+ \int_{\Omega} ((\text{id} + DW)^{-\top} \nabla u^W) \cdot ((\text{id} + DW)^{-\top} \nabla p^W) \det(\text{id} + DW) dx \\ &- \int_{\Omega} (f \circ T_W) p^W \det(\text{id} + DW) dx. \end{aligned}$$

Notice that  $\frac{\partial}{\partial W} \mathcal{L}(W, u^W, p^W)$  is the shape derivative  $J'(\Omega_W; \cdot)$ . Thus we find that the stationarity of  $\Omega_W$  is equivalent to the requirement that the solution of the nonlinear system

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & E^W & \cdot & E^W \\ \cdot & \cdot & \cdot & \cdot & -(N^W)^* \\ \cdot & \cdot & E^W & -N^W & \cdot \end{pmatrix} \begin{pmatrix} u^W \\ p^W \\ V_{\text{proj grad}}^W \\ F^W \\ \Pi^W \end{pmatrix} + \begin{pmatrix} \frac{\partial}{\partial u} \mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial p} \mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial W} \mathcal{L}(W, u^W, p^W) \\ \cdot \\ \cdot \end{pmatrix} = 0$$

satisfies  $V_{\text{proj grad}}^W = 0$ . In view of the injectivity of  $N^W$ , this is equivalent to  $F^W = 0$ . Here, “ $\cdot$ ” stands for a zero block. We mention that the first two equations in this system correspond to the adjoint and state equation on  $\Omega_W$  but pulled back to  $\Omega$ , respectively. Moreover, note that the solution  $(u^W, p^W, V_{\text{proj grad}}^W, F^W, \Pi^W)$  of the above system is the pull-back of the solution of the state equation, adjoint equation and the projected shape gradient of the system (5.6) formulated on the domain  $\Omega_W$ .

Together with the requirement that the deformation field  $W$  itself is induced by some normal force  $G$ , we have to solve the nonlinear system

$$(7.1) \quad \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \text{id} & \cdot \\ E & -N & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & E^W & \cdot & E^W \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -(N^W)^\star \\ \cdot & \cdot & \cdot & \cdot & E^W & -N^W & \cdot \end{pmatrix} \begin{pmatrix} W \\ G \\ u^W \\ p^W \\ V_{\text{proj grad}}^W \\ F^W \\ \Pi^W \end{pmatrix} + \begin{pmatrix} \cdot \\ \cdot \\ \frac{\partial}{\partial u} \mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial p} \mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial W} \mathcal{L}(W, u^W, p^W) \\ \cdot \\ \cdot \end{pmatrix} = 0.$$

As before,  $N$  and  $E$  denote the normal force operator and the elasticity operator on  $\Omega$ .

The system (7.1) for  $W$  and the further, auxiliary unknowns corresponds to the nonlinear system  $\Phi(x + \delta x) = 0$  for the step  $\delta x$ . For convenience, we recall the meaning of the seven equations in (7.1). The first equation requires  $F^W = 0$ , i.e., the stationarity of the updated domain  $\Omega_W$ . The second equation is the requirement that the displacement  $W$  is induced by the (normal) force  $G$ . The third and fourth equation are the adjoint and state equation on  $\Omega_W$ . Finally, the last three equations are the pull-back of the system (5.6) on  $\Omega_W$  to  $\Omega$ .

We can now describe a step of our Newton-like procedure for the solution of the nonlinear system (7.1). Suppose that  $\Omega$  is the current domain and consider an iterate of the form  $(0, 0, u, p, V_{\text{proj grad}}, F, \Pi)$  with the state, the adjoint state, and the solution of (5.6) on  $\Omega$ . Notice that for this iterate, the residual of (7.1) is  $(F, 0, 0, 0, 0, 0, 0)$ . Next we linearize the system (7.1) about this current iterate w.r.t. all seven variables. We refrain from stating the lengthy formula for the linear system which results. In practice, we generate this linear system governing the Newton step using the algorithmic differentiation capabilities of FENICS (Logg, Mardal, Wells, et al., 2012). From the solution of that linear system we only extract the Newton update for the perturbation field. We refer to it as  $W$  since its current value is zero. We then apply  $W$  to the current domain  $\Omega$  to obtain the new domain  $(\text{id} + W)(\Omega)$ . The six remaining variables are updated in a different fashion. Rather than using the solution from the Newton step, we solve again the state and adjoint state equations on the new domain, as well as the system (5.6) returning the projected shape gradient. This procedure can be understood as a Newton-like method with nonlinear updates for some of the variables. It ensures that the new iterate is of the same form as above. Moreover, it allows us access to the projected shape gradient and its norm in every iteration so that we can use  $\|V_{\text{proj grad}}\|_E \leq \varepsilon_{\text{tol}}$  as a stopping criterion as we did for the restricted gradient method.

Numerically, we have observed some instabilities if the current iterate  $\Omega$  is far from being stationary. Moreover we wish to establish a step size control in order to monitor the Armijo condition (5.18) and the mesh quality condition (5.19). To this



end we added a regularization term  $-G/\alpha$  to the first equation of (7.1), i.e., we obtain (7.2)

$$\begin{pmatrix} \cdot & -\alpha^{-1}\text{id} & \cdot & \cdot & \cdot & \text{id} & \cdot \\ E & -N & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & E_W & \cdot & E_W \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -N_W^* \\ \cdot & \cdot & \cdot & \cdot & E_W & -N_W & \cdot \end{pmatrix} \begin{pmatrix} W \\ G \\ u \\ p \\ V_{\text{proj grad}} \\ F \\ \Pi \end{pmatrix} + \begin{pmatrix} \cdot \\ \cdot \\ \frac{\partial}{\partial u} \mathcal{L}(W, u, p) \\ \frac{\partial}{\partial p} \mathcal{L}(W, u, p) \\ \frac{\partial}{\partial W} \mathcal{L}(W, u, p) \\ \cdot \\ \cdot \end{pmatrix} = 0.$$

Thus, the update resulting from the solution of the Newton system satisfies  $-\alpha^{-1}\delta G + \delta F = -F$ . Heuristically, this leads to  $\delta G \approx \alpha F$  for small  $\alpha$ . Consequently, the transformation field which is applied to the current domain  $\Omega$  satisfies  $W = E^{-1} N \delta G \approx \alpha V_{\text{proj grad}}$  is essentially a scaled (restricted) gradient direction for small  $\alpha$ . Therefore, similar as in a Levenberg–Marquardt method, we will refer to  $\alpha$  as the damping parameter and it serves the same purpose as the step length parameter in Algorithm 1.

A discrete variant of our Newton-like method is readily derived and given as Algorithm 2. In order to determine an appropriate damping parameter we consider analogues of the Armijo condition (5.18) and the mesh quality criterion (5.19). For the sake of clarity we re-state them with the relevant quantities for the Newton-like method. In particular, we use the step length  $\alpha = 1$  therein, since the scaling of the step is already realized by the damping in (7.2). The Armijo condition becomes

$$(7.3) \quad J_h((\text{id} + W_h)(\Omega_h)) \leq J_h(\Omega_h) + \sigma J'_h(\Omega_h; W_h).$$

with some parameter  $\sigma \in (0, 0.5)$ . The mesh quality criterion holds if

$$(7.4) \quad \frac{1}{2} \leq \det(\text{id} + DW_h) \leq 2, \quad \|DW_h\|_F \leq 0.3.$$

is satisfied in every cell. In addition we verify that  $W_h$  yields a descent direction. If any of the above conditions fails, we decrease the damping parameter  $\alpha$ .

**8. Numerical Results: Newton-Like Method.** This section is devoted to numerical results obtained by solving the same problem as in section 6 using the Newton-like method as described in section 7. As mentioned in section 6, our implementation is freely available, see Etling et al., 2018. For this approach the stopping criterion

$$(8.1) \quad \|V_{\text{proj grad}, h}\|_{E_h} \leq \varepsilon_{\text{tol}} = 10^{-9}$$

was satisfied after 12 iterations and 6 seconds on the previously used mesh with 469 vertices and 864 elements. In this case we used the line search parameters  $\beta = 0.1$  and  $\sigma = 0.1$  and an initial value of  $\alpha = 10^{-2}$ . Young's modulus and the Poisson ratio are  $E_0 = 1.0$  and  $\nu = 0.4$  as before. Some of the intermediate shapes are shown in Figure 7. As was already mentioned, we have the linear system in each Newton step assembled using the algorithmic differentiation capabilities of FENICS and solved in the same way as we did for the gradient method. In this scenario the geometry condition (7.4) led to a decrease in the damping parameter  $\alpha$  four times total (in iterations 3, 4, 5, and 7), while the Armijo condition (7.3) never necessitated a decrease in  $\alpha$ . We conjecture that the geometry condition (7.4) is triggered here more often compared to the gradient methods since the Newton-like method tends to produce

**Algorithm 2:** Restricted Newton method for (5.16).

---

**Data:** Initial domain  $\Omega_h$   
Initial damping parameter  $\alpha$ , convergence tolerance  $\varepsilon_{\text{tol}}$ ,  
line search parameters  $\beta \in (0, 1)$ ,  $\sigma \in (0, 0.5)$   
**Result:** Improved domain  $\Omega_h$  on which (5.16) holds up to  $\varepsilon_{\text{tol}}$

```

1 for  $i \leftarrow 1$  to  $\infty$  do
2   Solve the discrete state equation (4.2) for  $u_h$ ;
3   Solve the discrete adjoint equation (4.4) for  $p_h$ ;
4   Solve (5.12) for  $V_{\text{proj grad},h}$  with shape derivative  $J'_h(\Omega_h; \cdot)$  from (4.6);
5   if  $\langle E_h V_{\text{proj grad},h}, V_{\text{proj grad},h} \rangle \leq \varepsilon_{\text{tol}}^2$  then
6     STOP, the current iterate  $\Omega_h$  is almost stationary for (5.16);
7   end
8   Increase damping parameter  $\alpha \leftarrow \alpha/\beta$ ;
9   Solve the Newton system associated with (7.2) with damping parameter  $\alpha$ 
    and extract the first component as  $W_h$ ;
10  while  $J'_h(\Omega_h; W_h) \geq 0$  holds, or (7.3) or (7.4) is violated do
11    Decrease damping parameter  $\alpha \leftarrow \beta \alpha$ ;
12    Solve the Newton system associated with (7.2) with damping parameter
     $\alpha$  and extract the first component as  $W_h$ ;
13  end
14  Transform the domain according to  $\Omega_h \leftarrow (\text{id} + W_h)(\Omega_h)$ ;
15 end

```

---

Mesh Level		Restricted Gradient		Restricted Newton	
Vertices	Cells	Iter	Time [s]	Iter	Time [s]
127	216	527	10	9	3
469	864	864	38	11	7
1801	3456	1481	244	13	48
7057	13824	2353	1733	14	319

Table 1: Number of iterations and time of execution for the 2D example at different mesh levels to reach the tolerance (6.1) for the restricted gradient method. Moreover, for the restricted Newton method we used the stopping criterion (8.1) with a tolerance of  $10^{-8}$  and an initial damping parameter  $\alpha_0 = 10^7$ .

steps with larger norm  $\|W_h\|_{E_h}$ . However the perturbation of identity transformation imposes a limit on the step size, which leads to a reduction of  $\alpha$ . In the final iteration, the value of  $\alpha = 10^5$  is reached.

The experiments up to here were obtained on a coarse mesh with 469 vertices and 864 elements. We also studied the dependence of iteration numbers and CPU time on the mesh level, both for the restricted shape gradient and restricted Newton methods. Finer mesh levels are obtained by uniform refinement. Table 1 shows the size of the mesh in terms of the number of cells and vertices together with the number of iterations required for the convergence of both algorithms, and the time of execution.

Finally, we provide the results of the Newton-like method for an example in 3D. This time, the right hand side of the state equation in (2.1) is given by  $f(x, y, z) =$

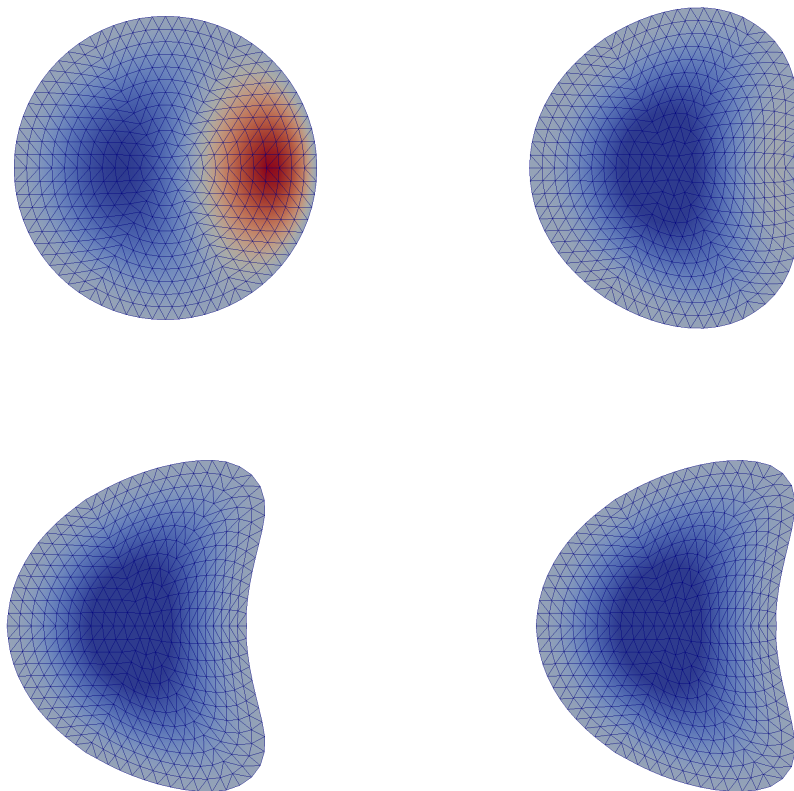


Fig. 7: Intermediate shapes  $\Omega_h$  obtained with the restricted Newton method at iterations 0, 4, 9, 12.

$2.5(x+0.4-y^2)^2+x^2+y^2+z^2-1$ . Moreover, the initial shape is a cube with 729 vertices and 3072 elements. All other data remains the same as above. Figure 8 shows the initial, some intermediate and the final shapes obtained using Algorithm 2 after 21 iterations and 286 seconds using the same tolerance as in the 2D case. Concerning the Armijo and geometry conditions (7.3) and (7.4) and the value of the damping parameter  $\alpha$ , we observe a very similar behavior as in the 2D case. The final value is  $\alpha = 10^6$ .

**9. Conclusions.** In this paper we introduce the concept of restricted mesh deformations for the computational solution of shape optimization problems involving PDEs. In a nutshell, we only admit perturbations fields which are induced by normal boundary forces. We argue that the stationarity condition (5.11) which does not impose any restriction on the mesh deformations leads to degenerate meshes and premature stopping. By contrast, we were able to solve the corresponding restricted stationarity condition (5.16) to high accuracy even with a gradient method. We also propose a Newton-like method based on restricted mesh deformations which exhibits

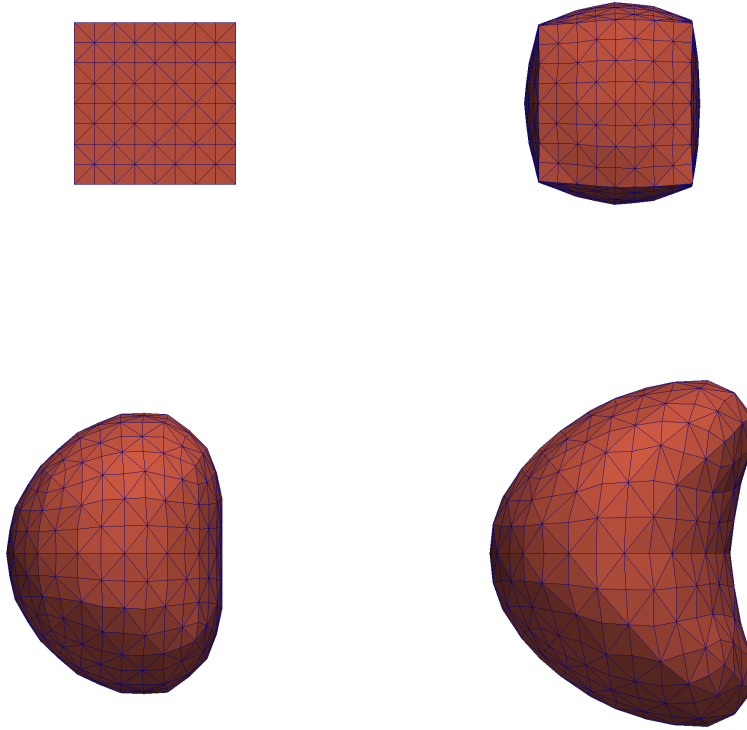


Fig. 8: Intermediate shapes  $\Omega_h$  obtained with the restricted Newton method at iterations 0, 7, 12, 21 for the 3D example.

fast convergence.

It is not clear whether (5.16) are the optimality conditions of a discrete optimization problem in Euclidean space. We conjecture that (5.16) are the optimality conditions for a problem defined on a discrete shape manifold, whose tangent space is represented by restricted mesh deformations.

**Acknowledgments.** The third author acknowledges support by the German Academic Exchange Service (DAAD) within the Doctoral Programm 2017/18.

#### References.

Adams, R.; Fournier, J. (2003). *Sobolev Spaces*. 2nd ed. New York: Academic Press.

- Attouch, H.; Buttazzo, G.; Michaille, G. (2014). *Variational analysis in Sobolev and BV spaces*. 2nd ed. Vol. 17. MOS-SIAM Series on Optimization. Applications to PDEs and optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, pp. xii+793. DOI: [10.1137/1.9781611973488](https://doi.org/10.1137/1.9781611973488).
- Bänsch, E.; Morin, P.; Nochetto, R. H. (2005). “A finite element method for surface diffusion: the parametric case”. *Journal of Computational Physics* 203.1, pp. 321–343. DOI: [10.1016/j.jcp.2004.08.022](https://doi.org/10.1016/j.jcp.2004.08.022).
- Berggren, M. (2010). “A unified discrete-continuous sensitivity analysis method for shape optimization”. *Applied and numerical partial differential equations*. Vol. 15. Computational Methods in Applied Sciences. Springer, New York, pp. 25–39. DOI: [10.1007/978-90-481-3239-3\\_4](https://doi.org/10.1007/978-90-481-3239-3_4).
- Bonito, A.; Nochetto, R. H.; Pauletti, M. S. (2010). “Geometrically consistent mesh modification”. *SIAM Journal on Numerical Analysis* 48.5, pp. 1877–1899. DOI: [10.1137/100781833](https://doi.org/10.1137/100781833).
- Delfour, M.; Payre, G.; Zolesio, J.-P. (1985). “An optimal triangulation for second-order elliptic problems”. *Computer Methods in Applied Mechanics and Engineering* 50, pp. 231–261. DOI: [10.1016/0045-7825\(85\)90095-7](https://doi.org/10.1016/0045-7825(85)90095-7).
- Delfour, M.; Zolésio, J.-P. (2011). *Shapes and Geometries*. 2nd ed. Society for Industrial and Applied Mathematics. DOI: [10.1137/1.9780898719826](https://doi.org/10.1137/1.9780898719826).
- Doğan, G.; Morin, P.; Nochetto, R. H.; Verani, M. (2007). “Discrete gradient flows for shape optimization and applications”. *Computer Methods in Applied Mechanics and Engineering* 196.37-40, pp. 3898–3914. DOI: [10.1016/j.cma.2006.10.046](https://doi.org/10.1016/j.cma.2006.10.046).
- Dokken, J. S.; Funke, S. W.; Johansson, A.; Schmidt, S. (2018). *Shape optimization using the finite element method on multiple meshes with Nitsche coupling*. SIAM Journal on Scientific Computing, under revision. arXiv: [1806.09821](https://arxiv.org/abs/1806.09821).
- Eppler, K.; Harbrecht, H. (2005). “A regularized Newton method in electrical impedance tomography using shape Hessian information”. *Control and Cybernetics* 34.1, pp. 203–225.
- Etling, T.; Herzog, R.; Loayza, E.; Wachsmuth, G. (2018). *First and Second Order Shape Optimization based on Restricted Mesh Deformations*. DOI: [10.5281/zenodo.2547481](https://doi.org/10.5281/zenodo.2547481).
- Feppon, F.; Allaire, G.; Bordeu, F.; Cortial, J.; Dapogny, C. (2018). *Shape Optimization of a Coupled Thermal Fluid-Structure Problem in a Level Set Mesh Evolution Framework*. HAL: [hal-01686770](https://hal.archives-ouvertes.fr/hal-01686770).
- Gangl, P.; Langer, U.; Laurain, A.; Meftahi, H.; Sturm, K. (2015). *Shape Optimization of an Electric Motor subject to Nonlinear Magnetostatics*. Tech. rep. eprint: [arXiv:1501.04752](https://arxiv.org/abs/1501.04752).
- Giacomini, M.; Pantz, O.; Trabelsi, K. (2017). “Certified descent algorithm for shape optimization driven by fully-computable *a posteriori* error estimators”. *ESAIM. Control, Optimisation and Calculus of Variations* 23.3, pp. 977–1001. DOI: [10.1051/cocv/2016021](https://doi.org/10.1051/cocv/2016021).
- Haslinger, J.; Mäkinen, R. (2003). *Introduction to Shape Optimization*. Philadelphia: SIAM.
- Hiptmair, R.; Paganini, A.; Sargheini, S. (2015). “Comparison of approximate shape gradients”. *BIT. Numerical Mathematics* 55.2, pp. 459–485. DOI: [10.1007/s10543-014-0515-z](https://doi.org/10.1007/s10543-014-0515-z).
- Iglesias, J. A.; Sturm, K.; Wechsung, F. (2017). *Shape optimisation with nearly conformal transformations*. Tech. rep. eprint: [arXiv:1710.06496](https://arxiv.org/abs/1710.06496).

- Ito, K.; Kunisch, K.; Peichl, G. H. (2008). “Variational approach to shape derivatives”. *ESAIM. Control, Optimisation and Calculus of Variations* 14.3, pp. 517–539. DOI: [10.1051/cocv:2008002](https://doi.org/10.1051/cocv:2008002).
- Laurain, A.; Sturm, K. (2016). “Distributed shape derivative via averaged adjoint method and applications”. *ESAIM. Mathematical Modelling and Numerical Analysis* 50.4, pp. 1241–1267. DOI: [10.1051/m2an/2015075](https://doi.org/10.1051/m2an/2015075).
- Li, X. S. (2005). “An overview of SuperLU: algorithms, implementation, and user interface”. *Association for Computing Machinery. Transactions on Mathematical Software* 31.3, pp. 302–325. DOI: [10.1145/1089014.1089017](https://doi.org/10.1145/1089014.1089017).
- Logg, A.; Mardal, K.-A.; Wells, G. N., et al. (2012). *Automated Solution of Differential Equations by the Finite Element Method*. Springer. DOI: [10.1007/978-3-642-23099-8](https://doi.org/10.1007/978-3-642-23099-8).
- Luenberger, D. G. (1969). *Optimization by Vector Space Methods*. John Wiley.
- Mignot, F. (1976). “Contrôle dans les inéquations variationnelles elliptiques”. *Journal of Functional Analysis* 22.2, pp. 130–185.
- Morin, P.; Nochetto, R. H.; Pauletti, M. S.; Verani, M. (2012). “Adaptive finite element method for shape optimization”. *ESAIM. Control, Optimisation and Calculus of Variations* 18.4, pp. 1122–1149. DOI: [10.1051/cocv/2011192](https://doi.org/10.1051/cocv/2011192).
- Novruzzi, A.; Roche, J. R. (2000). “Newton’s method in shape optimisation: a three-dimensional case”. *BIT. Numerical Mathematics* 40.1, pp. 102–120. DOI: [10.1023/A:1022370419231](https://doi.org/10.1023/A:1022370419231).
- Roth, R.; Ulbrich, S. (2013). “A Discrete Adjoint Approach for the Optimization of Unsteady Turbulent Flows”. *Flow, Turbulence and Combustion* 90.4, pp. 763–783. DOI: [10.1007/s10494-012-9439-3](https://doi.org/10.1007/s10494-012-9439-3).
- Schmidt, S.; Ilic, C.; Schulz, V.; Gauger, N. R. (2011). “Airfoil design for compressible inviscid flow based on shape calculus”. *Optimization and Engineering. International Multidisciplinary Journal to Promote Optimization Theory & Applications in Engineering Sciences* 12.3, pp. 349–369. DOI: [10.1007/s11081-011-9145-3](https://doi.org/10.1007/s11081-011-9145-3).
- Schmidt, S.; Schulz, V. H. (2009). “Impulse response approximations of discrete shape Hessians with application in CFD”. *SIAM Journal on Control and Optimization* 48.4, pp. 2562–2580. DOI: [10.1137/080719844](https://doi.org/10.1137/080719844).
- Schmidt, S.; Schulz, V. H. (2010). “Shape derivatives for general objective functions and the incompressible Navier-Stokes equations”. *Control and Cybernetics* 39.3, pp. 677–713.
- Schmidt, S.; Schulz, V.; Ilic, C.; Gauger, N. (2011). “Three dimensional large scale aerodynamic shape optimization based on shape calculus”. *41st AIAA Fluid Dynamics Conference and Exhibit*. American Institute of Aeronautics and Astronautics. DOI: [10.2514/6.2011-3718](https://doi.org/10.2514/6.2011-3718).
- Schneider, R.; Jimack, P. K. (2008). “On the evaluation of finite element sensitivities to nodal coordinates”. *Electronic Transactions on Numerical Analysis* 32, pp. 134–144.
- Schulz, V. H. (2014). “A Riemannian view on shape optimization”. *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics* 14.3, pp. 483–501. DOI: [10.1007/s10208-014-9200-5](https://doi.org/10.1007/s10208-014-9200-5).
- Schulz, V. H.; Siebenborn, M. (2016). “Computational comparison of surface metrics for PDE constrained shape optimization”. *Computational Methods in Applied Mathematics* 16.3, pp. 485–496. DOI: [10.1515/cmam-2016-0009](https://doi.org/10.1515/cmam-2016-0009).
- Schulz, V. H.; Siebenborn, M.; Welker, K. (2015). “Structured inverse modeling in parabolic diffusion problems”. *SIAM Journal on Control and Optimization* 53.6, pp. 3319–3338. DOI: [10.1137/140985883](https://doi.org/10.1137/140985883).

- Schulz, V. H.; Siebenborn, M.; Welker, K. (2016). “Efficient PDE constrained shape optimization based on Steklov-Poincaré type metrics”. *SIAM Journal on Optimization* 26.4, pp. 2800–2819. DOI: [10.1137/15M1029369](https://doi.org/10.1137/15M1029369).
- Sokołowski, J.; Zolésio, J.-P. (1992). *Introduction to Shape Optimization*. New York: Springer.
- Sturm, K. (2015). “Minimax Lagrangian Approach to the Differentiability of Nonlinear PDE Constrained Shape Functions without Saddle Point Assumptions”. *SIAM Journal on Control and Optimization* 53.4, pp. 2017–2039. DOI: [10.1137/130930807](https://doi.org/10.1137/130930807).
- Sturm, K. (2016). “Shape optimization with nonsmooth cost functions: from theory to numerics”. *SIAM Journal on Control and Optimization* 54.6, pp. 3319–3346. DOI: [10.1137/16M1069882](https://doi.org/10.1137/16M1069882).
- Udawalpola, R.; Berggren, M. (2008). “Optimization of an acoustic horn with respect to efficiency and directivity”. *International Journal for Numerical Methods in Engineering* 73.11, pp. 1571–1606. DOI: [10.1002/nme.2132](https://doi.org/10.1002/nme.2132).
- Wachsmuth, G. (2016). “Towards M-stationarity for optimal control of the obstacle problem with control constraints”. *SIAM Journal on Control and Optimization* 54.2, pp. 964–986. DOI: [10.1137/140980582](https://doi.org/10.1137/140980582).
- Wilke, D. N.; Kok, S.; Groenwold, A. A. (2005). “A quadratically convergent unstructured remeshing strategy for shape optimization”. *International Journal for Numerical Methods in Engineering* 65.1, pp. 1–17. DOI: [10.1002/nme.1430](https://doi.org/10.1002/nme.1430).
- Zunino, P. (2004). “Multidimensional Pharmacokinetic Models Applied to the Design of Drug-Eluting Stents”. *Cardiovascular Engineering* 4.2, pp. 181–191. DOI: [10.1023/b:care.0000031547.39178.cb](https://doi.org/10.1023/b:care.0000031547.39178.cb).