SIAM J. SCI. COMPUT.
Vol. 42, No. 2, pp. A1200–A1225

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

TOMMY ETLING†, ROLAND HERZOG†, ESTEFANÍA LOAYZA†, AND GERD WACHSMUTH‡

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

DOI. 10.1137/19M1241465

1. Introduction.

Shape optimization is ubiquitous in the design of structures of all kinds, from drug eluting stents [46] to aircraft wings [31] or horn-like structures appearing in devices for acoustic or electromagnetic waves [43]. 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:

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

Here $u(\Omega)$ is the solution of the underlying PDE defined on the domain Ω, 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 of the current domain Ω. The perturbations are carried out in terms of either the

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 29, 2019; accepted for publication (in revised form) February 25, 2020; published electronically April 16, 2020.

https://doi.org/10.1137/19M1241465

Funding: This work was partially supported by DFG through grants HE 6077/10-1 and WA 3636/4-1. The work of the third author was supported by DAAD through Doctoral Programme 2017/18.

†Technische Universität Chemnitz, Faculty of Mathematics, Professorship Numerical Mathematics (Partial Differential Equations), 09107 Chemnitz, Germany (tommy.etling.tu-chemnitz.de, https://www.tu-chemnitz.de/mathematik/part_dgl/people/etling, roland.herzog@mathematik.tu-chemnitz.de, https://www.tu-chemnitz.de/herzog, estefania.loayza@mathematik.tu-chemnitz.de, https://www.tu-chemnitz.de/mathematik/part_dgl/people/loayza).

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

perturbation of identity or the velocity method. We refer the reader to [8, 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 [40, Theorem 2.27]. For volume expressions, we refer the reader, for instance, to [20, 17]. 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 Ω; see, e.g., [31, 36, 33, 32]. More sophisticated techniques include quasi-Newton or Hessian-based inner products; see [11, 27, 37, 35]. This perturbation field is then used to update the domain Ω inside a line search method, where the transformed domain

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

associated with the step size α is obtained from the perturbation of identity approach. This class of computational schemes equally applies on the continuous and discrete levels. In the latter case, the domain Ω is often represented by a computational mesh.

A widely used alternative approach, after discretization, is to parametrize the possible displacements of the boundary nodes only. The movements of the interior nodes then follow as a second step as the result of some possibly nonlinear map in response to the boundary node displacements. As above, the latter can be obtained utilizing either the volume or the boundary expressions of the shape derivative. We refer the reader to [29, 30, 23, 5] for examples of this strategy.

In any case, while the computation of the shape derivative is based on either 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 Ω 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 Ω 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. [9], 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. For 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, [45, 26, 42, 10, 13]. 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.

[3, 9] describe several techniques such as mesh regularization, space adaptivity, and 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, [26] 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 proposed in [6]. [15] address the issue of spurious descent directions, attributed to discretization errors in the underlying PDE model, via a goal-oriented mesh adaptation approach. Recently, [18] 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 [37, 38, 36], who advocate the linear elasticity model as the inner product to convert shape derivative into a shape gradient. In particular in [38] 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., a very small norm of the restricted gradient. Both gradient and Newton schemes in two and three dimensions are considered. An implementation in the finite element software FEniCS is available as open-source on GitHub; see [12].

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:

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

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,

$$\text{Find } u \in H^1_0(\Omega) \qquad \text{such that} \qquad \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x = \int_\Omega f\, v \,\mathrm{d}x \quad \forall v \in H^1_0(\Omega). \tag{2.2}$$

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 to quasi-open sets, quasi-continuity, quasi-everywhere (q.e.), and related notions, we refer the reader to [2, section 5.8]. We consider the slightly relaxed problem

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

Let us recall that $H^1_0(\Omega) = \{u \in H^1_0(\mathbb{R}^d) \mid u = 0 \text{ q.e. in } \mathbb{R}^d \setminus \Omega\}$ and $H^{-1}(\Omega)$ is the dual space of $H^1_0(\Omega)$. The PDE in (2.3) is also to be understood in the weak sense, i.e.,

$$\text{Find } u \in H^1_0(\Omega) \qquad \text{such that} \qquad \int_D \nabla u \cdot \nabla v = \int_D f\, v \,\mathrm{d}x \quad \forall v \in H^1_0(\Omega).$$

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 $u \le 0$ in (2.3). Indeed, if (Ω, 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 chosen to be quasi-continuous. This pair is again admissible due to

$$\int_D \nabla \min(u, 0) \cdot \nabla v \,\mathrm{d}x = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x = \int_D f\, v \,\mathrm{d}x \qquad \forall v \in H^1_0(\{u < 0\}),$$

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 (Ω, u).

Now, let $\{(\Omega_n, u_n)\}$ be a minimizing sequence for (2.3) with $u_n \le 0$ 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, $u \le 0$. 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 $\hat u \le u$ 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 $-u \ge v \ge 0$. For $v_n := \min(-u_n, v)$ we have $v_n \in H_0^1(\Omega_n)$ due to $v \ge 0$. Moreover, $v_n \rightharpoonup \min(-u, v) = v$ in $H_0^1(D)$; see [44, Lemma 4.1]. Thus,

$$\begin{aligned}\int_D f\, v \,\mathrm{d}x &= \lim_{n\to\infty} \int_D f\, v_n \,\mathrm{d}x = \lim_{n\to\infty} \int_D \nabla u_n \cdot \nabla v_n \,\mathrm{d}x \\ &= \lim_{n\to\infty} \int_D \nabla(u_n + v)\cdot\nabla(v_n - v) + \nabla u_n \cdot \nabla v - \nabla v \cdot \nabla(v_n - v)\,\mathrm{d}x \\ &= \lim_{n\to\infty} \int_D -\nabla|\min(-u_n - v, 0)|^2 + \nabla u_n \cdot \nabla v \,\mathrm{d}x \le \int_D \nabla u \cdot \nabla v \,\mathrm{d}x.\end{aligned}$$

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

$$\int_{\hat\Omega} \nabla(\hat u - u)\cdot\nabla v \,\mathrm{d}x \le 0 \qquad \forall v \in H_0^1(D) \text{ satisfying } -u \ge v \ge 0.$$

Now, we can use a density argument (see [25, Lemme 3.4]) to obtain that this inequality holds for all $v \in H_0^1(\hat\Omega)$ which satisfy $v \ge 0$. Using $v = \max(\hat u - u, 0)$ implies $\max(\hat u - u, 0) = 0$, i.e., $\hat u \le u$. Finally, the optimality of $(\hat\Omega, \hat u)$ follows from

$$\int_D \hat u \,\mathrm{d}x \le \int_D u \,\mathrm{d}x = \lim_{n\to\infty} \int_D u_n \,\mathrm{d}x.$$

∎

Remark 2.2. There is a deeper reason for $\hat u \le u$ 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 capacitary measure μ. We refer to [2, section 5.8.4] for a nice introduction to capacitary measures. Due to $u \le 0$ we have (in a certain sense) $\mu\, u \le 0$ and therefore $\hat u \le u$ follows from the maximum principle since "$-\Delta \hat u = f \le f - \mu\, u = -\Delta u$." However, we included the above direct proof because it does not rely on the notion of capacitary 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

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

if the integrand j satisfies

(2.4a) $\quad j(x,\cdot)$ is monotonically increasing on $(-\infty, 0]$ and nonnegative on $[0,\infty)$,

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

(2.4c) $\quad u_n \rightharpoonup u$ in $H_0^1(D)$ implies $\displaystyle\int_D j(u)\,\mathrm{d}x \le \liminf_{n\to\infty} \int_D j(u_n)\,\mathrm{d}x$.

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

$$\begin{aligned}\int_{\hat\Omega} j(\cdot,\hat u)\,\mathrm{d}x \le \int_{\hat\Omega} j(\cdot,u)\,\mathrm{d}x &= \int_D j(\cdot,u)-j(\cdot,0)\,\mathrm{d}x + \int_{\{u<0\}} j(0)\,\mathrm{d}x\\ &\le \liminf_{n\to\infty}\int_D j(\cdot,u_n)-j(\cdot,0)\,\mathrm{d}x + \int_{\{u_n<0\}} j(0)\,\mathrm{d}x\\ &= \liminf_{n\to\infty}\int_{\Omega_n} j(\cdot,u_n)\,\mathrm{d}x.\end{aligned}$$

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

$$\int_{\{u<0\}} j(0)\,\mathrm{d}x \le \liminf_{n\to\infty}\int_{\{u_n<0\}} j(0)\,\mathrm{d}x.$$

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 a rather standard problem we will be able to directly apply results from [19]. 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 ∂D and $\partial\Omega$, respectively. Moreover we assume $\overline{\Omega}\subset D$ so that Ω has a positive distance to the boundary of D.

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

$$T_\alpha = \mathrm{id} + \alpha V, \tag{3.1}$$

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_α is invertible for all $\alpha\in[0,\tau]$.

By a straightforward application of [19, 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*

$$\begin{aligned}J'(\Omega;V) = &\int_\Omega u\,(\operatorname{div} V)\,\mathrm{d}x\\ &+\int_\Omega (\nabla u)^\top\left[(\operatorname{div} V)\,\mathrm{id} - DV - DV^\top\right]\nabla p\,\mathrm{d}x - \int_\Omega \operatorname{div}(f\,V)\,p\,\mathrm{d}x,\end{aligned} \tag{3.2}$$

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

$$\text{Find } p\in H_0^1(\Omega) \quad \text{such that} \quad \int_D \nabla p\cdot\nabla v\,\mathrm{d}x = -\int_D v\,\mathrm{d}x \quad \forall\ v\in H_0^1(\Omega). \tag{3.3}$$

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 [8, Chapter 9, Theorem 3.6]. We state it here in a particularized version for problem (2.1). From now on, ν denotes the outer unit normal vector along the boundary $\partial\Omega$ of Ω.

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

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

Notice that under the assumption that Ω 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, [1, 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 [41], [40, Chapter 3.3], [16, Example 3.3]. We also refer the reader to [17], 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 Ω_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,

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

defined over an approximation Ω_h of Ω 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

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

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

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

We refer the reader to [14, 42, 38, 36] for examples of this procedure.

Let us denote by $J_h(\Omega_h)$ the reduced objective value in (4.3), i.e., $J_h(\Omega_h) = \int_{\Omega_h} u_h \,\mathrm{d}x$, 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,

$$\text{Find } p_h \in S_0^1(\Omega_h) \quad \text{such that} \quad \int_{\Omega_h} \nabla p_h \cdot \nabla v_h \,\mathrm{d}x = -\int_{\Omega_h} v_h \,\mathrm{d}x \quad \forall\ v_h \in S_0^1(\Omega_h). \tag{4.4}$$

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

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

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*

$$J_h'(\Omega_h; V_h) = \int_{\Omega_h} u_h \,(\operatorname{div} V_h)\,\mathrm{d}x + \int_{\Omega_h} (\nabla u_h)^\top \left[(\operatorname{div} V_h)\,\mathrm{id} - DV_h - DV_h^\top\right] \nabla p_h \,\mathrm{d}x - \int_{\Omega_h} \operatorname{div}(f\,V_h)\,p_h \,\mathrm{d}x. \tag{4.6}$$

The proof of this theorem follows along the lines of the continuous case; see, e.g., [17, 20]. A detailed derivation can be found in [7, 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 interelement contributions; see also the discussion in [4].
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 Ω_h is addressed by differentiating the finite element matrices. We refer to [34, 4, 28] for examples of this procedure.

Despite the simplicity of obtaining 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 Ω_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 Ω is unknown. We begin with the computational mesh Ω_h shown in Figure 1 (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 1.

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

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

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 2. At this point, finite element computations break down.

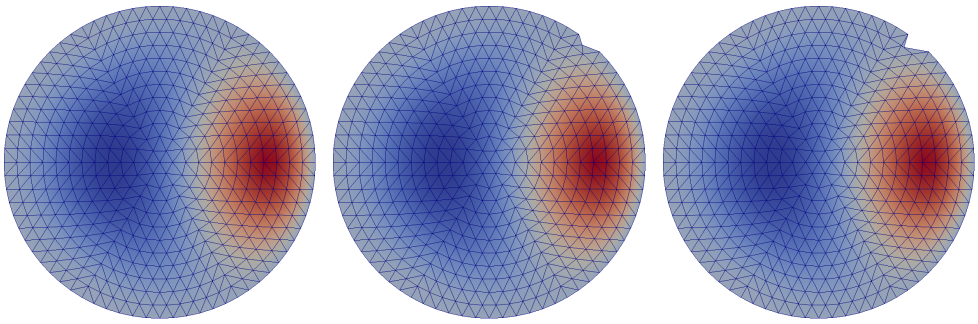

FIG. 1. *Evolution of the mesh under* $\alpha \mapsto T_\alpha(\Omega_h)$ *with perturbation field* V_h *given in Example* 4.3 *at* $\alpha = 0.00$, $\alpha = 0.05$, $\alpha = 0.10$ *(from left to right). The solutions* u_h *of the state equation* (4.2) *are also shown.*

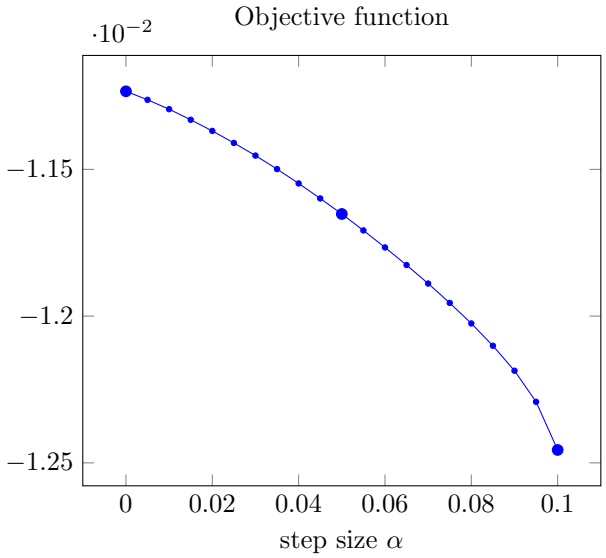

FIG. 2. $\alpha \mapsto J\big(T_\alpha(\Omega_h)\big)$ *for Example* 4.3. *The step sizes* $\alpha = 0.00$, $\alpha = 0.05$, $\alpha = 0.10$ *belonging to the domains in Figure* 1 *are highlighted.*

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.

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 [8, note, 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 alloy 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 $(W^{1,\infty}(\Omega)^d)^\star$. In order to utilize this information for moving the domain Ω, we have to convert this dual element into a proper function. We follow the approach of [38]. To this end, we introduce the elasticity operator $E : H^1(\Omega)^d \to (H^1(\Omega)^d)^\star$ via

$$\langle E\,V,\,W\rangle := \int_\Omega 2\,\mu\,\varepsilon(V) : \varepsilon(W) + \lambda\,\operatorname{trace}(\varepsilon(V))\,\operatorname{trace}(\varepsilon(W)) + \delta\,V\cdot W\,\mathrm{d}x \tag{5.1}$$

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, μ, λ 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 semidefinite 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., $\langle E\,V,\,V\rangle \geq \underline{c}\,\|V\|^2_{H^1(\Omega)^d}$ with some $\underline{c} > 0$. This result is due to Korn's inequality; see, for instance, [2, Proposition 6.6.1]. Thus, E is an isomorphism and it furnishes $H^1(\Omega)^d$ with an inner product $(V,\,W)_E := \langle E\,V,\,W\rangle$ 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 $J'(\Omega;\cdot) \in (H^1(\Omega)^d)^\star$. This holds, e.g., if Ω 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

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

The solution of this problem yields the negative shape gradient

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

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

$$\langle NF,\,V\rangle = \int_{\partial\Omega} F\,(V\cdot\nu)\,\mathrm{d}s \tag{5.4}$$

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 $J'(\Omega;\cdot) = N g_\Omega$ 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

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

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, [24, Chapter 9.3, Theorem 1]. We therefore obtain the following necessary and sufficient optimality conditions for (5.5) in saddle-point form:

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

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

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

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

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 $\mathrm{im}(E^{-1}N)$. 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 $V_{\mathrm{grad}} = -E^{-1}J'(\Omega;\cdot)$.

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

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

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

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 Finite element space $S_0^1(\Omega_h) \subset H_0^1(\Omega_h)$, we recall from (4.5) the discrete space of mesh deformations

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

and the boundary space

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

We recall that the discrete shape derivative $J_h'(\Omega_h;\cdot) \in (S^1(\Omega_h)^d)^\star$ was given in (4.6). Moreover, the discretization directly leads to the discretized operators $E_h : S^1(\Omega_h)^d \to (S^1(\Omega_h)^d)^\star$, $N_h : S^1(\partial\Omega_h) \to (S^1(\Omega_h)^d)^\star$ which are defined via

$$\langle E_h\, V_h,\, W_h\rangle := \int_{\Omega_h} 2\,\mu\,\boldsymbol{\varepsilon}(V_h) : \boldsymbol{\varepsilon}(W_h) + \lambda\ \mathrm{trace}(\boldsymbol{\varepsilon}(V_h))\ \mathrm{trace}(\boldsymbol{\varepsilon}(W_h)) + \delta\, V_h \cdot W_h \,\mathrm{d}x,$$
$$\langle N_h\, F_h,\, V_h\rangle := \int_{\partial\Omega_h} F_h\,(V_h \cdot \nu)\,\mathrm{d}s$$

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

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

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

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 [8]. 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

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

as a discretization of the continuous problem (1.1).

Hence, we consider the discretization of (5.5)

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

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

$$\begin{pmatrix} E_h & 0 & E_h \\ 0 & 0 & -N_h^\star \\ E_h & -N_h & 0 \end{pmatrix} \begin{pmatrix} V_h \\ F_h \\ \Pi_h \end{pmatrix} = \begin{pmatrix} -J_h'(\Omega_h;\cdot) \\ 0 \\ 0 \end{pmatrix}. \tag{5.13}$$

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

$$\begin{aligned} \langle E_h\, V_{\mathrm{proj\,grad},h},\, V_{\mathrm{proj\,grad},h}\rangle &= -\langle E_h\, V_{\mathrm{proj\,grad},h},\, \Pi_h\rangle - J_h'(\Omega_h; V_{\mathrm{proj\,grad},h}) \\ &= -\langle N_h\, F_h,\, \Pi_h\rangle - J_h'(\Omega_h; V_{\mathrm{proj\,grad},h}) \\ &= -J_h'(\Omega_h; V_{\mathrm{proj\,grad},h}) \end{aligned} \tag{5.14}$$

since $N_h^\star \, \Pi_h = 0$ 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

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

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),

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

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 Ω_h (together with a line search).

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

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

This, in turn, is equivalent to

$$\int_{\partial\Omega_h} F_h \, (V_{\text{grad},h} \cdot \nu) \, \mathrm{d}s = 0 \qquad \forall F_h \in S^1(\partial\Omega_h). \tag{5.17}$$

This means that Ω_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 Ω_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 a backtracking strategy to comply with the Armijo condition

$$J_h\big((\mathrm{id} + \alpha \, V_{\text{proj grad},h})(\Omega_h)\big) \leq J_h(\Omega_h) + \sigma \, \alpha \, J_h'(\Omega_h; V_{\text{proj grad},h}). \tag{5.18}$$

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

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

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

Algorithm 1: Restricted gradient method for (5.16).

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

for $i \leftarrow 1$ **to** ∞ **do**
  Solve the discrete state equation (4.2) for u_h;
  Solve the discrete adjoint equation (4.4) for p_h;
  Solve (5.12) for $V_{\mathrm{proj\,grad},h}$ with shape derivative $J'(\Omega_h;\cdot)$ from (4.6);
  if $\langle E_h V_{\mathrm{proj\,grad},h},\, V_{\mathrm{proj\,grad},h}\rangle \le \varepsilon_{\mathrm{tol}}^2$ **then**
    STOP, the current iterate Ω_h is almost stationary for (5.16);
  end
  Increase step size $\alpha \leftarrow \alpha/\beta$;
  while (5.18) *or* (5.19) *is violated* **do**
    Decrease step size $\alpha \leftarrow \beta\,\alpha$;
  end
  Transform the domain according to $\Omega_h \leftarrow (\mathrm{id} + \alpha\, V_{\mathrm{proj\,grad},h})(\Omega_h)$;
end

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

$$\langle E_h V_{\mathrm{proj\,grad},h},\, V_{\mathrm{proj\,grad},h}\rangle = -J_h'(\Omega_h; V_{\mathrm{proj\,grad},h}) \le \varepsilon_{\mathrm{tol}}^2 \tag{5.20}$$

as a convergence criterion for some small $\varepsilon_{\mathrm{tol}} > 0$. These considerations lead to Algorithm 1.

5.3. Comparison with the approach of [38]. In [38], 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.,

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

they propose that "Only test functions whose support includes Γ_{int} are considered [...]"; see [38, p. 2813]. In their problem formulation, Γ_{int} corresponds to $\partial\Omega_h$ in our formulation. We interpret this as follows. We denote by $D_h : S^1(\Omega_h)^d \to S^1(\Omega_h)^d$ the projection operator defined via

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

for all nodes x from the mesh Ω_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 $V_{h,\mathrm{SSW}}$ is computed via the solution of

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

As confirmed by [39], this interpretation coincides with the implementation from [38]. We compare this suggestion with our approach.

- The deformation $V_{h,\mathrm{SSW}}$ can be computed faster than $V_{\mathrm{proj\,grad},h}$, since the linear system is smaller than (5.13).
- The deformation $V_{h,\mathrm{SSW}}$ 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, $V_{h,\mathrm{SSW}}$ may fail to be a descent direction, since

$$J_h'(\Omega_h, V_{h,\mathrm{SSW}}) = -J_h'\big(\Omega_h, E_h^{-1} J_h'(\Omega_h, D_h\,\cdot)\big)$$

  might be positive. This indeed does happen in our numerical examples; see section 6 below. In contrast, our suggestion $V_{\mathrm{proj\,grad},h}$ 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 $V_{h,\mathrm{SSW}}$ 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.

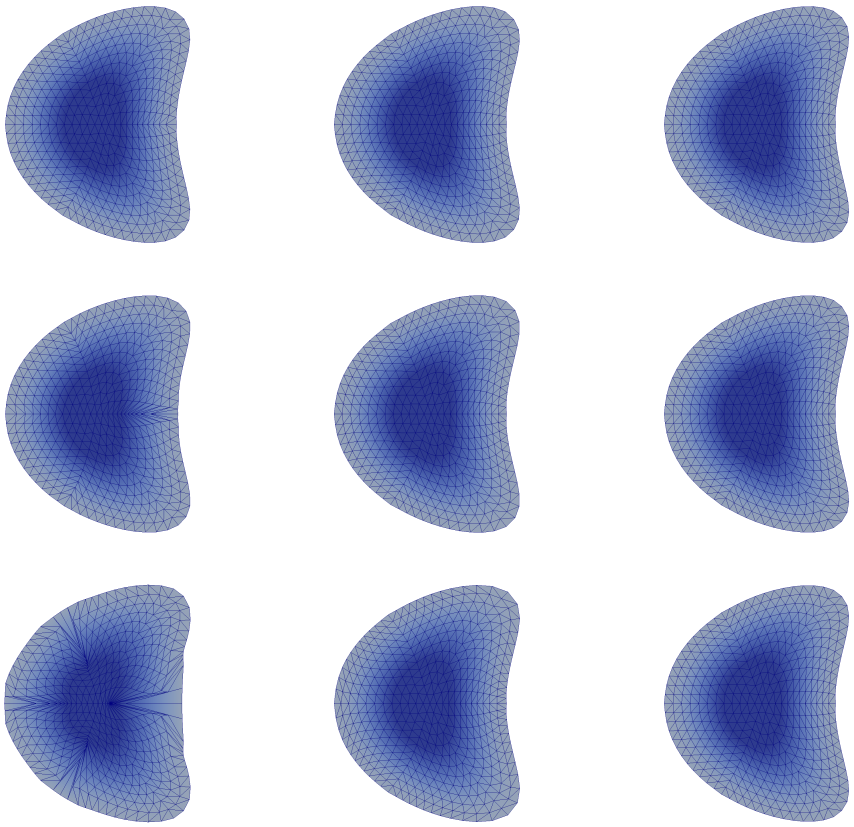

FIG. 3. *Intermediate shapes Ω_h obtained with the classical gradient method (left), the gradient method from* [38] *(middle), and the restricted gradient method (right) at iterations* 300, 600, *and* 1500, 1435, *and* 864, *respectively.*

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 $V_{\mathrm{proj\,grad},h}$ is replaced everywhere by the negative shape gradient $V_{\mathrm{grad},h}$ from (5.10). In addition, we also compare it to a method utilizing the deformation fields $V_{h,\mathrm{SSW}}$ obtained along the lines of [38]. We refer to the latter as a gradient-like method since it may fail to produce descent directions. Consequently, $V_{h,\mathrm{SSW}}$ 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 1 (left). For this first result, the mesh has 864 triangles and 469 vertices but computations on refined meshes are reported later in Table 1.

We implemented the restricted gradient method, Algorithm 1, its classical counterpart, as well as the gradient-like method from [38] in FEniCS, version 2018.1 [22]. We report computational results obtained on a machine with an Intel Xeon CPU E5–4640 at 2.4 GHz.

Our implementation is freely available on GitHub; see [12]. All derivatives were automatically generated by the built-in algorithmic differentiation capabilities of FEniCS. The restricted shape gradient $V_{\mathrm{proj\,grad},h}$, 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 [21], i.e., with the setting `use_umfpack = False`.

The restricted gradient method reached the desired tolerance

$$\|V_{\mathrm{proj\,grad},h}\|_{E_h} = \sqrt{|J_h'(\Omega_h; V_{\mathrm{proj\,grad},h})|} \le \varepsilon_{\mathrm{tol}} = 10^{-7} \tag{6.1}$$

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

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

On the other hand, the gradient-like method from [38] 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

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

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 Ω_h during the iteration of all three methods for comparison. It can clearly be inferred that the initial iterates are virtually identical.

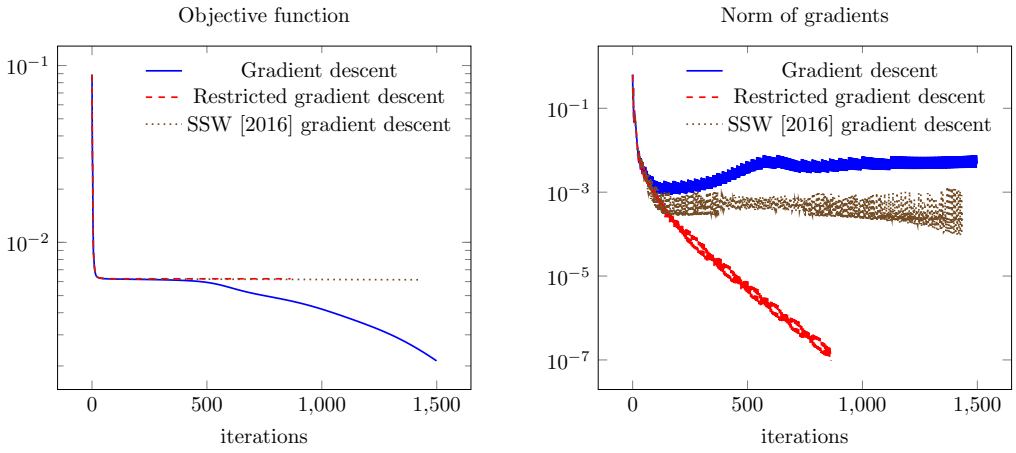

FIG. 4. *History of the objective value* $J_h(\Omega_h) + 0.1$ *(left) and the norm of the gradients* $\|V_{\mathrm{grad},h}\|_{E_h}$ *(for the gradient descent method),* $\|V_{\mathrm{proj\,grad},h}\|_{E_h}$ *(for the restricted gradient method), and* $\sqrt{J_h'(\Omega_h; V_{h,\mathrm{SSW}})}$ *(for the gradient-like method from* [38]*) along the iterations (right).*

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

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

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 [38] 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 [38]. 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 [38] are equally good. However, the latter eventually failed to produce descent directions in 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.

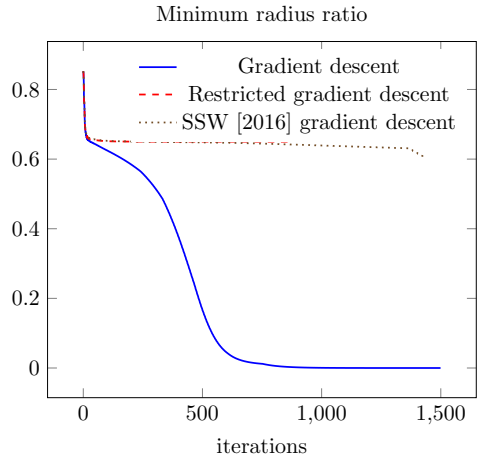

FIG. 5. *History of the mesh quality indicator (minimum over all cells of two times the inradius divided by the circumradius of the cell).*

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 $V_{\mathrm{grad},h}$, i.e., the solution of (5.10), is nonzero and in fact essentially the same for iterations 300, 600, and 864 shown. However, $V_{\mathrm{grad},h}$ 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 $\|V_{\mathrm{proj\,grad},h}\|_{E_h}$, as shown in 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 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 afterward. Let Ω 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 Ω is denoted by $(V_{\mathrm{proj\,grad}}, F, \Pi)$. Recall that our goal is to achieve $V_{\mathrm{proj\,grad}} = 0$ or, equivalently, $F = 0$; cf. (5.16). In practice, we impose a stopping criterion of the form $\|V_{\mathrm{proj\,grad}}\|_E \leq \varepsilon_{\mathrm{tol}}$ 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 a parallel with Newton's method for $\Phi(x) = 0$ for some $\Phi : \mathbb{R}^n \to \mathbb{R}^n$. We consider the equation $\Phi(x + \delta x) = 0$ for the unknown update δx. In our context the iterate x represents the current domain Ω and the update corresponds to a perturbation field W. Since the update takes Ω into a new domain, we need to manipulate the expression $\Phi(x + \delta x) = 0$ and pull it back to Ω. Finally, we linearize about $\delta x = 0$, which amounts to $\Phi(x) + D\Phi(x)\,\delta x = 0$.

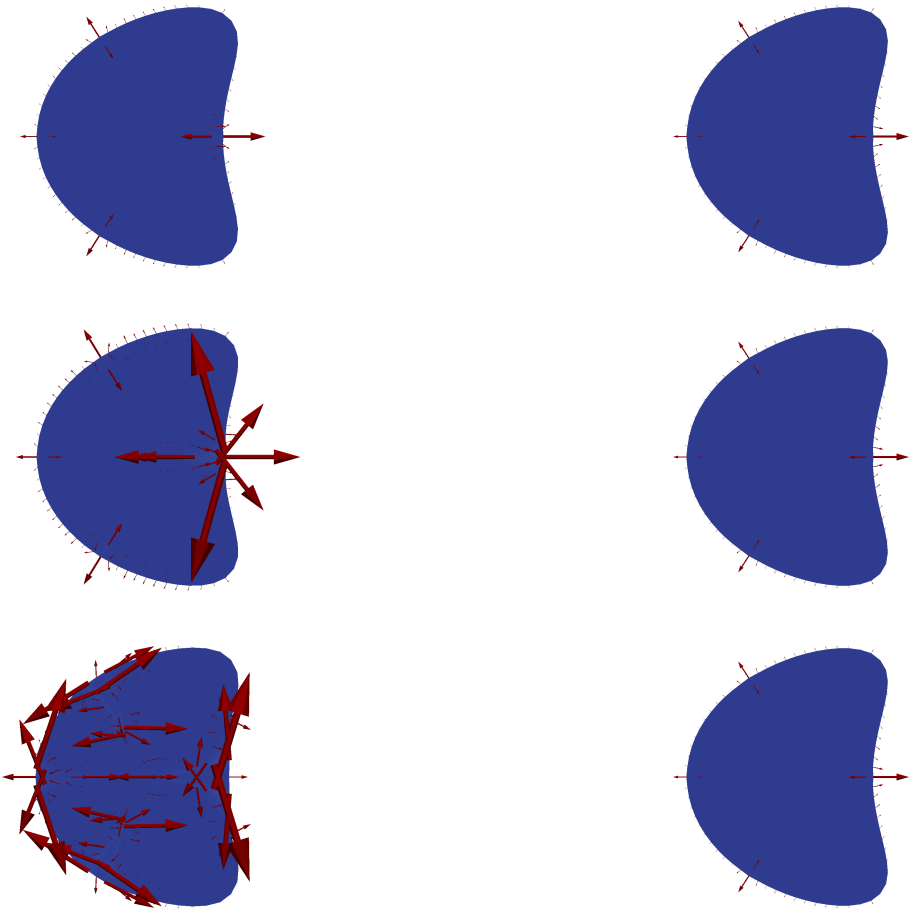

FIG. 6. *Visualization of the shape derivatives* $J_h'(\Omega_h;\cdot)$ *obtained with the classical gradient method (left) and the restricted gradient method (right) at iterations* 300, 600, 1500 *(left) and* 300, 600, 864 *(right), respectively.*

In our Newton method we seek a deformation field W (taking the role of δx above) such that the updated domain $\Omega_W := (\mathrm{id}+W)(\Omega)$ is stationary in the sense that the solution of (5.6) (at Ω_W instead of Ω) satisfies $V^W_{\text{proj grad}} = 0$. 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 Ω_W, we introduce the elasticity operator $E_W : H^1(\Omega_W)^d \to (H^1(\Omega_W)^d)^\star$ and the normal force operator $N_W : L^2(\partial\Omega_W) \to (H^1(\Omega_W)^d)^\star$ on Ω_W analogously to (5.1) and (5.4). With the transformation field $T_W := \mathrm{id}+W : \Omega \to \Omega_W$, we define the pull-backs $E^W : H^1(\Omega)^d \to (H^1(\Omega)^d)^\star$ of E_W and $N^W : L^2(\partial\Omega) \to (H^1(\Omega)^d)^\star$ of N_W via

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

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

Since we wish to achieve conditions defined on the current domain Ω, rather than on the unknown transformed domain Ω_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 Ω. Using the usual integral substitution and chain rule, and denoting the pulled-back solutions of the state and adjoint equations on Ω_W by u^W and p^W, respectively, we obtain $\mathcal{L} : W^{1,\infty}(\Omega)^d \times H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$, defined as

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

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 Ω_W is equivalent to the requirement that the solution of the nonlinear system

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

satisfies $V^W_{\mathrm{proj\,grad}} = 0$. 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 Ω_W but pulled back to Ω, respectively. Moreover, note that the solution $(u^W, p^W, V^W_{\mathrm{proj\,grad}}, F^W, \Pi^W)$ of the above system is the pull-back of the solution of the state equation, the adjoint equation, and the projected shape gradient of the system (5.6) formulated on the domain Ω_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)
$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \mathrm{id} & \cdot \\ E & -N & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & E^W & \cdot & E^W \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -(N^W)^\star \\ \cdot & \cdot & \cdot & \cdot & E^W & -N^W & \cdot \end{pmatrix} \begin{pmatrix} W \\ G \\ u^W \\ p^W \\ V^W_{\mathrm{proj\,grad}} \\ F^W \\ \Pi^W \end{pmatrix} + \begin{pmatrix} \cdot \\ \cdot \\ \frac{\partial}{\partial u}\mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial p}\mathcal{L}(W, u^W, p^W) \\ \frac{\partial}{\partial W}\mathcal{L}(W, u^W, p^W) \\ \cdot \\ \cdot \end{pmatrix} = 0.$$

As before, N and E denote the normal force operator and the elasticity operator on Ω.

The system (7.1) for W and the further, auxiliary unknowns corresponds to the nonlinear system $\Phi(x + \delta x) = 0$ for the step δx. 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 Ω_W. The second equation is the requirement

that the displacement W is induced by the (normal) force G. The third and fourth equations are the adjoint and state equation on Ω_W. Finally, the last three equations are the pull-back of the system (5.6) on Ω_W to Ω.

We can now describe a step of our Newton-like procedure for the solution of the nonlinear system (7.1). Suppose that Ω 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 Ω. 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 [22]. 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 Ω to obtain the new domain $(\mathrm{id}+W)(\Omega)$. 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 Ω 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)
$$\begin{pmatrix} \cdot & -\alpha^{-1}\,\mathrm{id} & \cdot & \cdot & \cdot & \mathrm{id} & \cdot \\ E & -N & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & E_W & \cdot & E_W \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -N_W^\star \\ \cdot & \cdot & \cdot & \cdot & E_W & -N_W & \cdot \end{pmatrix} \begin{pmatrix} W \\ G \\ u \\ p \\ V_{\text{proj grad}} \\ F \\ \Pi \end{pmatrix} + \begin{pmatrix} \cdot \\ \cdot \\ \frac{\partial}{\partial u}\mathcal{L}(W,u,p) \\ \frac{\partial}{\partial p}\mathcal{L}(W,u,p) \\ \frac{\partial}{\partial W}\mathcal{L}(W,u,p) \\ \cdot \\ \cdot \end{pmatrix} = 0.$$

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 α. Consequently, the transformation field which is applied to the current domain Ω satisfies $W = E^{-1} N \delta G \approx \alpha V_{\text{proj grad}}$ is essentially a scaled (restricted) gradient direction for small α. Therefore, similarly as in the Levenberg–Marquardt method, we will refer to α 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 restate 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

$$J_h\big((\mathrm{id}+W_h)(\Omega_h)\big) \leq J_h(\Omega_h) + \sigma\, J_h'(\Omega_h; W_h) \tag{7.3}$$

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

Algorithm 2: Restricted Newton method for (5.16).

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

```
for i ← 1 to ∞ do
    Solve the discrete state equation (4.2) for u_h;
    Solve the discrete adjoint equation (4.4) for p_h;
    Solve (5.12) for V_{proj grad,h} with shape derivative J_h'(Ω_h; ·) from (4.6);
    if ⟨E_h V_{proj grad,h}, V_{proj grad,h}⟩ ≤ ε_tol^2 then
        STOP, the current iterate Ω_h is almost stationary for (5.16);
    end
    Increase damping parameter α ← α/β;
    Solve the Newton system associated with (7.2) with damping parameter
     α and extract the first component as W_h;
    while J_h'(Ω_h; W_h) ≥ 0 holds, or (7.3) or (7.4) is violated do
        Decrease damping parameter α ← β α;
        Solve the Newton system associated with (7.2) with damping
         parameter α and extract the first component as W_h;
    end
    Transform the domain according to Ω_h ← (id + W_h)(Ω_h);
end
```

$$\frac{1}{2} \leq \det(\mathrm{id} + DW_h) \leq 2, \qquad \|DW_h\|_F \leq 0.3, \tag{7.4}$$

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 α.

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 [12]. For this approach the stopping criterion

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

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 α four times total (in iterations 3, 4, 5, and 7), while the Armijo condition (7.3) never necessitated a decrease in α. 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 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 α. In the final iteration, the value of $\alpha = 10^5$ is reached.

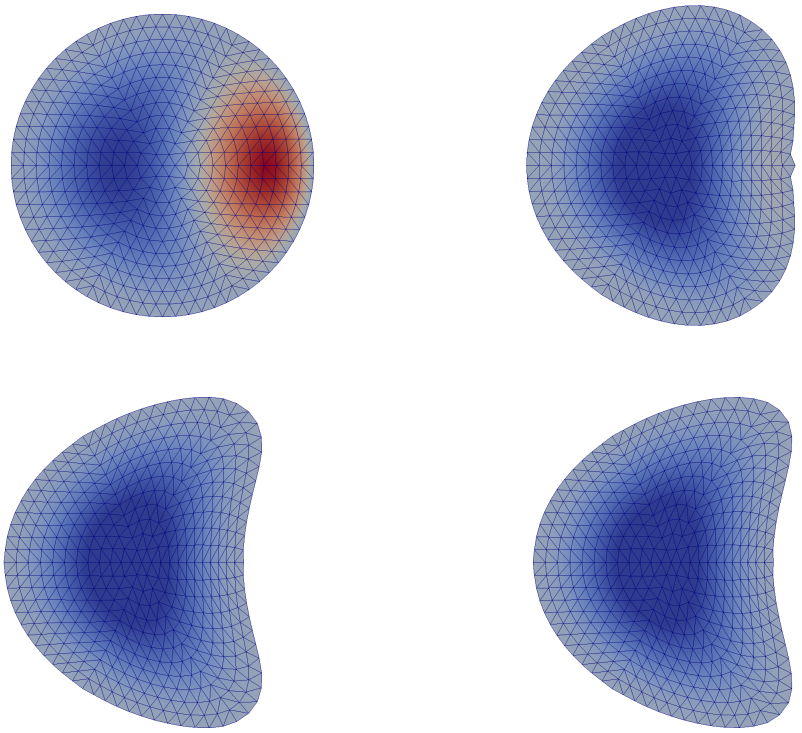

FIG. 7. *Intermediate shapes Ω_h obtained with the restricted Newton method at iterations* 0, 4, 9, 12.

TABLE 1

Number of iterations and time of execution for the two-dimensional (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* $\alpha_0 = 10^7$.

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

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, for both 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 three dimensions. This time, the right-hand side of the state equation in (2.1) is given by $f(x, y, z) = 2.5\,(x + 0.4 - y^2)^2 + x^2 + y^2 + z^2 - 1$. Moreover, the initial shape is a cube with 729 vertices and 3072 elements. All other data remains the same as

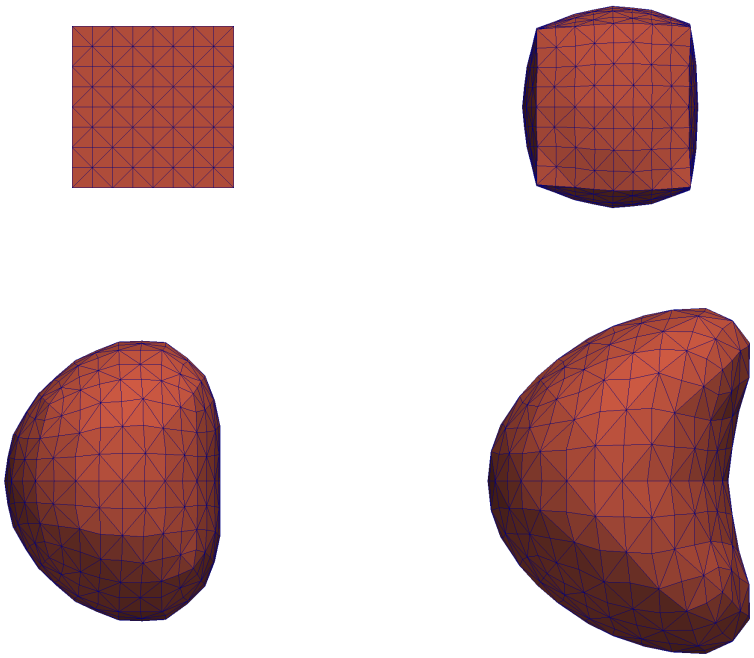

FIG. 8. *Intermediate shapes* Ω_h *obtained with the restricted Newton method at iterations* 0, 7, 12, 21 *for the* 3*D example.*

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 α, 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 perturbation 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 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.

REFERENCES

[1] R. Adams and J. Fournier, *Sobolev Spaces*, 2nd ed. Academic Press, New York, 2003.

[2] H. Attouch, G. Buttazzo, and G. Michaille, *Variational Analysis in Sobolev and BV Spaces*: *Applications to PDEs and Optimization,* 2nd ed., MOS-SIAM Ser. Optim. 17, SIAM, Philadelphia, 2014, https://doi.org/10.1137/1.9781611973488.

[3] E. Bänsch, P. Morin, and R. H. Nochetto, *A finite element method for surface diffusion: The parametric case*, J. Comput. Phys., 203 (2005), pp. 321–343, https://doi.org/10.1016/j.jcp.2004.08.022.

[4] M. Berggren, *A unified discrete-continuous sensitivity analysis method for shape optimization*, in Applied and Numerical Partial Differential Equations, Comput. Methods Appl. Sci. 15, Springer, New York, 2010, pp. 25–39, https://doi.org/10.1007/978-90-481-3239-3_4.

[5] K. Bobrowski, E. Ferrer, E. Valero, and H. Barnewitz, *Aerodynamic shape optimization using geometry surrogates and adjoint method*, AIAA J., 55 (2017), pp. 3304–3317, https://doi.org/10.2514/1.J055766.

[6] A. Bonito, R. H. Nochetto, and M. S. Pauletti, *Geometrically consistent mesh modification*, SIAM J. Numer. Anal., 48 (2010), pp. 1877–1899, https://doi.org/10.1137/100781833.

[7] M. Delfour, G. Payre, and J.-P. Zolesio, *An optimal triangulation for second-order elliptic problems*, Comput. Methods Appl. Mech. Engrg., 50 (1985), pp. 231–261, https://doi.org/10.1016/0045-7825(85)90095-7.

[8] M. Delfour and J.-P. Zolésio, *Shapes and Geometries*, 2nd ed. Adv. Des. Control 22, SIAM, Philadelphia, 2011, https://doi.org/10.1137/1.9780898719826.

[9] G. Doğan, P. Morin, R. H. Nochetto, and M. Verani, *Discrete gradient flows for shape optimization and applications*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 3898–3914, https://doi.org/10.1016/j.cma.2006.10.046.

[10] J. r. S. Dokken, S. W. Funke, A. Johansson, and S. Schmidt, *Shape optimization using the finite element method on multiple meshes with Nitsche coupling*, SIAM J. Sci. Comput., 41 (2019), pp. A1923–A1948, https://doi.org/10.1137/18M1189208.

[11] K. Eppler and H. Harbrecht, *A regularized Newton method in electrical impedance tomography using shape Hessian information*, Control Cybernet., 34 (2005), pp. 203–225.

[12] T. Etling, R. Herzog, E. Loayza, and G. Wachsmuth, *First and Second Order Shape Optimization Based on Restricted Mesh Deformations*, https://doi.org/10.5281/zenodo.2547481, 2018.

[13] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, and C. Dapogny, *Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework*, SEMA J., 76 (2019), pp. 413–458, https://doi.org/10.1007/s40324-018-00185-4.

[14] P. Gangl, U. Langer, A. Laurain, H. Meftahi, and K. Sturm, *Shape optimization of an electric motor subject to nonlinear magnetostatics*, SIAM J. Sci. Comput., 37 (2015), pp. B1002–B1025, https://doi.org/10.1137/15100477X.

[15] M. Giacomini, O. Pantz, and K. Trabelsi, *Certified descent algorithm for shape optimization driven by fully-computable a posteriori error estimators*, ESAIM Control Optim. Calc. Vari., 23 (2017), pp. 977–1001, https://doi.org/10.1051/cocv/2016021.

[16] J. Haslinger and R. Mäkinen, *Introduction to Shape Optimization*, Adv. Des. Control 7, SIAM, Philadelphia, 2003.

[17] R. Hiptmair, A. Paganini, and S. Sargheini, *Comparison of approximate shape gradients*, BIT, 55 (2015), pp. 459–485, https://doi.org/10.1007/s10543-014-0515-z.

[18] J. A. Iglesias, K. Sturm, and F. Wechsung, *Two-dimensional shape optimization with nearly conformal transformations*, SIAM J. Sci. Comput., 40 (2018), pp. A3807–A3830, https://doi.org/10.1137/17M1152711.

[19] K. Ito, K. Kunisch, and G. H. Peichl, *Variational approach to shape derivatives*, ESAIM Control Optim. Calc. Var., 14 (2008), pp. 517–539, https://doi.org/10.1051/cocv:2008002.

[20] A. Laurain and K. Sturm, *Distributed shape derivative via averaged adjoint method and applications*, ESAIM Math. Model. Numer. Anal., 50 (2016), pp. 1241–1267, https://doi.org/10.1051/m2an/2015075.

[21] X. S. Li, *An overview of SuperLU: Algorithms, implementation, and user interface*, Trans. Math. Softw., 31 (2005), pp. 302–325, https://doi.org/10.1145/1089014.1089017.

[22] A. Logg, K.-A. Mardal, and G. N. Wells, *Automated Solution of Differential Equations by the Finite Element Method*, Lect. Notes Comput. Sci. Eng. 84, Springer, New York, 2012, https://doi.org/10.1007/978-3-642-23099-8.

[23] C. Lozano, *On mesh sensitivities and boundary formulas for discrete adjoint-based gradients in inviscid aerodynamic shape optimization*, J. Comput. Phys., 346 (2017), pp. 403–436, https://doi.org/10.1016/j.jcp.2017.06.025.

[24] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley, New York, 1969.
[25] F. Mignot, *Contrôle dans les inéquations variationelles elliptiques*, J. Funct. Anal., 22 (1976), pp. 130–185.
[26] P. Morin, R. H. Nochetto, M. S. Pauletti, and M. Verani, *Adaptive finite element method for shape optimization*, ESAIM Control Optim. Calc. Var., 18 (2012), pp. 1122–1149, https://doi.org/10.1051/cocv/2011192.
[27] A. Novruzi and J. R. Roche, *Newton's method in shape optimisation: A three-dimensional case*, BIT, 40 (2000), pp. 102–120, https://doi.org/10.1023/A:1022370419231.
[28] R. Roth and S. Ulbrich, *A discrete adjoint approach for the optimization of unsteady turbulent flows*, Flow Turbulence Combustion, 90 (2013), pp. 763–783, https://doi.org/10.1007/s10494-012-9439-3.
[29] S. Schmidt, C. Ilic, V. Schulz, and N. R. Gauger, *Airfoil design for compressible inviscid flow based on shape calculus*, Optim. Eng., 12 (2011), pp. 349–369, https://doi.org/10.1007/s11081-011-9145-3.
[30] S. Schmidt, C. Ilic, V. Schulz, and N. R. Gauger, *Three-dimensional large-scale aerodynamic shape optimization based on shape calculus*, AIAA J., 51 (2013), pp. 2615–2627, https://doi.org/10.2514/1.j052245.
[31] S. Schmidt, V. Schulz, C. Ilic, and N. Gauger, *Three dimensional large scale aerodynamic shape optimization based on shape calculus*, in Proceedings of the 41st AIAA Fluid Dynamics Conference and Exhibit, American Institute of Aeronautics and Astronautics, 2011, https://doi.org/10.2514/6.2011-3718.
[32] S. Schmidt and V. H. Schulz, *Impulse response approximations of discrete shape Hessians with application in CFD*, SIAM J. Control Optim., 48 (2009), pp. 2562–2580, https://doi.org/10.1137/080719844.
[33] S. Schmidt and V. H. Schulz, *Shape derivatives for general objective functions and the incompressible Navier-Stokes equations*, Control Cybernet., 39 (2010), pp. 677–713.
[34] R. Schneider and P. K. Jimack, *On the evaluation of finite element sensitivities to nodal coordinates*, Electron. Trans. Numer. Anal., 32 (2008), pp. 134–144.
[35] V. H. Schulz, *A Riemannian view on shape optimization*, Found. Comput. Math., 14 (2014), pp. 483–501, https://doi.org/10.1007/s10208-014-9200-5.
[36] V. H. Schulz and M. Siebenborn, *Computational comparison of surface metrics for PDE constrained shape optimization*, Comput. Methods Appl. Math., 16 (2016), pp. 485–496, https://doi.org/10.1515/cmam-2016-0009.
[37] V. H. Schulz, M. Siebenborn, and K. Welker, *Structured inverse modeling in parabolic diffusion problems*, SIAM J. Control Optim., 53 (2015), pp. 3319–3338, https://doi.org/10.1137/140985883.
[38] V. H. Schulz, M. Siebenborn, and K. Welker, *Efficient PDE constrained shape optimization based on Steklov-Poincaré type metrics*, SIAM J. Optim., 26 (2016), pp. 2800–2819, https://doi.org/10.1137/15M1029369.
[39] M. Siebenborn, *Private communication*, 2020.
[40] J. Sokołowski and J.-P. Zolésio, *Introduction to Shape Optimization*, Springer, New York, 1992.
[41] K. Sturm, *Minimax Lagrangian approach to the differentiability of nonlinear PDE constrained shape functions without saddle point assumptions*, SIAM J. Control Optim., 53 (2015), pp. 2017–2039, https://doi.org/10.1137/130930807.
[42] K. Sturm, *Shape optimization with nonsmooth cost functions: from theory to numerics*, SIAM J. Control Optim., 54 (2016), pp. 3319–3346, https://doi.org/10.1137/16M1069882.
[43] R. Udawalpola and M. Berggren, *Optimization of an acoustic horn with respect to efficiency and directivity*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 1571–1606, https://doi.org/10.1002/nme.2132.
[44] G. Wachsmuth, *Towards M-stationarity for optimal control of the obstacle problem with control constraints*, SIAM J. Control Optim., 54 (2016), pp. 964–986, https://doi.org/10.1137/140980582.
[45] D. N. Wilke, S. Kok, and A. A. Groenwold, *A quadratically convergent unstructured remeshing strategy for shape optimization*, Internat. J. Numer. Methods Engrg., 65 (2005), pp. 1–17, https://doi.org/10.1002/nme.1430.
[46] P. Zunino, *Multidimensional pharmacokinetic models applied to the design of drug-eluting stents*, Cardiovascular Engrg., 4 (2004), pp. 181–191, https://doi.org/10.1023/b:care.0000031547.39178.cb.