SIAM J. CONTROL OPTIM.
Vol. 61, No. 4, pp. 2358–2378

A LINEAR VIEW ON SHAPE OPTIMIZATION*

STEPHAN SCHMIDT† AND VOLKER H. SCHULZ†

Abstract. Shapes do not define a linear space. This paper explores the linear structure of deformations as a representation of shapes. This transforms shape optimization to a variant of optimal control. The numerical challenges of this point of view are highlighted and a linear version of the second shape derivative is employed, leading to particular algorithms of shape Newton type.

Key words. shape optimization, PDE constrained optimization, numerical methods

MSC codes. 49M15, 49M41, 49Q10

DOI. 10.1137/22M1488910

1. Introduction. Shape optimization is a research topic of high interest and is also used in numerous fields of application. Here, we mention only some fundamental references [41, 10, 29, 20] and some selected applications [16, 36, 5]. As mentioned, e.g., in [10], shapes do not define a linear space. For example, there is no straight forward way to define the sum of two shapes. But linear spaces are the realm of standard optimization techniques—whether it be finite dimensional or infinite dimensional vector spaces. There are various ways to deal with this situation: enforcing finite dimensional parameterization of shapes like in CAD (computer aided design), which limits the variability of shapes; exploring the manifold structure of shapes like in [37, 44], which is restricted to certain shape classes; and the method of mapping [30, 24, 7], which maps the boundary of a straight reference domain to the deforming boundary under consideration. The paper [15] treats the special case of star-shaped domains where the circle serves as a reference domain.

Here, we investigate the space of deformations of an initial shape. These deformations are of type $id + V$, where V is a vector field to be determined. A standard optimization method like steepest decent or Newton applied directly to V results after n steps in $id + \sum_{k=1}^{n} V^k$, where V^k denote the increments defined by the respective method. In the context of shapes, this is called a linear representation of shapes in [45]. The problem with this way of application of optimization methods is the fact that there is no easily implemented way to guarantee that the resulting deformation $id + \sum_{k=1}^{n} V^k$ is invertible. Lack of invertibility typically leads to overlapping or intersecting computational meshes in a PDE (partial differential equation) constrained framework, where computational domain meshes have to be used and often cannot be replaced by meshes only on the shape boundary. After lack of invertibility arises in the iterations, no further progress is possible and the algorithm has to be aborted. That is the reason why this approach is usually not applied and numerically replaced by the Micheletti framework [28], [9, Chapter 3], where the iteration result after n steps may rather be written as $(id+V_n)\circ\cdots\circ(id+V_1)$. In [45], also the resulting group structure for this shape representation is investigated. In [43] the same framework is

*Received by the editors April 6, 2022; accepted for publication (in revised form) April 18, 2023; published electronically July 28, 2023.
https://doi.org/10.1137/22M1488910
Funding: Both authors received support from the Deutsche Forschungsgemeinschaft within Priority Program SPP 1962, "Non-smooth and Complementarity-based Distributed Parameter Systems: Simulation and Hierarchical Optimization," and the second author also within Research Training Group RTG 2126, "Algorithmic Optimization."

†Trier University, 54296 Trier, Germany (s.schmidt@uni-trier.de, volker.schulz@uni-trier.de).

used, where shape Hessians in domain and boundary formulations are investigated. We focus on domain formulations. The invertibility of each factor $(id + V_k)$ in this concatenation can be controlled in each optimization step, e.g., by step damping, which leads to invertibility of the product. Another computational advantage is the fact that the computation of the optimization steps in the Micheletti framework only need the local coordinates of the current shape and do not require the formation of the completely concatenated deformation. However, at first glance, the Micheletti framework does not seem to be amenable to the standard convergence theory of numerical optimization algorithms. In this paper, we observe that optimization steps carried out in the Micheletti framework are the same steps as if they are carried out in the naive vector space framework; just the coordinate representation is different. Thus, local asymptotic convergence properties for optimization methods can be carried over to shape optimization from optimization in standard vector spaces. This observation is also illustrated in the numerical results.

The disadvantage, however, of characterizing shapes by domain deformations is the lack of uniqueness of the optimal solutions, which poses challenges to numerical algorithms, remedies of which are also investigated in this paper. While this paper is closely connected to the perturbation of identity concept in shape calculus, it is related to the paper [4], which provides an optimal control representation of shapes which is different from shape calculus. However, in our paper, we use the standard shape calculus and show the relation to standard vector space algorithms, leading to a novel interpretation of shape Newton methods. There is a significantly large literature on shape Newton methods like [25, 14, 34] and several articles cited above. However, the novelty in our paper is the embedding in an overall linear Newton scheme.

This paper is organized as follows. Section 2 recalls the classical definitions of first and second order shape derivatives. Our main results are in section 3, where we show that the classical shape derivatives possess a linear space interpretation. Section 4 introduces a convenient way to derive necessary optimality conditions in weak form. A model problem is introduced in section 5, for which numerical results are given in section 6.

2. Shape derivatives. Here, we recall basic definitions of first and second shape derivatives. Let $d \in \mathbb{N}$ and $\tau > 0$. We denote by $\Omega \subset \mathbb{R}^d$ a bounded domain with Lipschitz boundary $\Gamma := \partial\Omega$ and by J a real-valued functional depending on it. Moreover, let $\{T_t\}_{t\in[0,\tau]}$ be a family of bijective mappings $T_t \colon \Omega \to \mathbb{R}^d$ such that $T_0 = id$. This family transforms the domain Ω into new perturbed domains $\Omega_t := T_t(\Omega) = \{T_t(x) \colon x \in \Omega\}$ with $\Omega_0 = \Omega$ and the boundary Γ into new perturbed boundaries $\Gamma_t := T_t(\Gamma) = \{T_t(x) \colon x \in \Gamma\}$ with $\Gamma_0 = \Gamma$. If you consider the domain Ω as a collection of material particles, which are changing their position in the time-interval $[0, \tau]$, then the family $\{T_t\}_{t\in[0,\tau]}$ describes the motion of each particle, i.e., at the time $t \in [0, \tau]$ a material particle $x \in \Omega$ has the new position $x_t := T_t(x) \in \Omega_t$ with $x_0 = x$. The motion of each such particle x could be described by the *velocity method,* i.e., as the flow $T_t(x) := \xi(t, x)$ determined by the initial value problem

$$\begin{aligned} \frac{d\xi(t,x)}{dt} &= V(\xi(t,x)), \\ \xi(0,x) &= x, \end{aligned} \tag{2.1}$$

or by the *perturbation of identity,* which is defined by $T_t(x) := x + tV(x)$, where V denotes a sufficiently smooth vector field. We will use the perturbation of identity throughout the paper. The *Eulerian derivative* of J at Ω in direction V is defined by

$$dJ(\Omega)[V] := \lim_{t\to 0^+} \frac{J(\Omega_t) - J(\Omega)}{t}. \tag{2.2}$$

The expression $dJ(\Omega)[V]$ is called the *shape derivative* of J at Ω in direction V and J *shape differentiable* at Ω if for all directions V the Eulerian derivative (2.2) exists and the mapping $V \mapsto dJ(\Omega)[V]$ is linear and continuous. For a thorough introduction to shape calculus, we refer to the monographs [10, 41]. In particular, [39] states that shape derivatives can always be expressed as boundary integrals under appropriate regularity conditions due to the Hadamard structure theorem. The shape derivative arises in two equivalent notational forms:

$$dJ_\Omega[V] := \int_\Omega F(x)V(x)\,dx \qquad \text{(domain formulation)}, \tag{2.3}$$

$$dJ_\Gamma[V] := \int_\Gamma f(s)V(s)^\top n(s)\,ds \qquad \text{(boundary formulation)}, \tag{2.4}$$

where $F(x)$ is a (differential) operator acting linearly on the perturbation vector field V and $f\colon \Gamma \to \mathbb{R}$ with

$$dJ_\Omega[V] = dJ(\Omega)[V] = dJ_\Gamma[V]. \tag{2.5}$$

The boundary formulation (2.4), $dJ_\Gamma[V]$, acting on the normal component of V has led to the interpretation as a tangential vector of a corresponding shape manifold in [37].

The second shape derivative is defined as the first shape derivative of a shape derivative. That necessitates a second flow movement after the first flow movement. If we denote a second flow induced by a vector field W by T_s^W, then the second shape derivative is defined as

$$d^2J[V,W] := \lim_{s\to 0^+} \frac{1}{s}\left(\lim_{t\to 0^+} \frac{J(T_s^W(\Omega_t) - J(T_s^W(\Omega))}{t} - \lim_{t\to 0^+} \frac{J(\Omega_t) - J(\Omega)}{t}\right).$$

In [10] concrete expressions for the second shape Hessian are discussed in the case of the velocity method.

In addition to these basic definitions, usually also concepts like local shape derivative and material derivative are needed in order to simplify the shape calculus. We use that in the sections below and refer to [6] for the specific usage in computations.

3. Shape optimization algorithms and convergence in deformation vector space. A standard shape optimization algorithm applies the same shape deformation T_t used in the definition of the shape derivative in order to generate iterated shapes via

$$\Omega^{k+1} = T_{t^k}^{V^k}(\Omega^k),$$

where V^k is related to the shape derivative $dJ(\Omega^k)[W]$. The earliest—to the knowledge of the authors—convergence result for this type of algorithm can be found in [22]. There, all V^k are assumed to be from a Hilbert space $\mathcal{H}$, the first and second order shape derivatives are assumed to exist, the second shape derivative is assumed to be uniformly bounded, and the step vector field V^k is defined by

$$b_k(V^k, W) = -dJ(\Omega^k)[W] \;\; \forall W \in \mathcal{H}, \tag{3.1}$$

where $b_k(.,.)$ are assumed to be uniformly lower and upper bounded bilinear forms in $\mathcal{H}$. If the step length t^k is chosen according to a backtracking line search, and J is

assumed bounded from below, then [22, Theorem 3.1] shows $\lim_{k\to\infty} J(\Omega^k) = J^*$ for some $J^* \in \mathbb{R}$ and $\lim_{k\to\infty} \|V^k\| = 0$.

This is a very general convergence result. However, it only states converge in the objective function and does not guarantee the existence of a limiting shape. Nor does it quantify the convergence speed. The key problem for further discussions is the proper notion of the distance in shape spaces. In [37], shapes are characterized as elements of a Riemannian manifold and the concepts of optimization on manifolds are carried over to shape manifolds. Within this framework, convergence analysis for steepest descent, Newton, and quasi-Newton methods is presented. The major drawback with this approach is, however, the requirement of C^∞ smoothness of the shapes under investigation, which does not really fit into a typical finite element framework from standard PDE numerics.

Here, we consider only shapes which result from deformations of an initial shape and identify the resulting shapes with the deformation mapping producing them. This is related to the concept of pre-shapes as discussed in detail in [26, 27], but we do not want to complicate the discussions unnecessarily with this concept. Starting from an initial shape Ω^0, here we assume to work only with deformed shapes of the type $T(\Omega^0) = \{T(x) : x \in \Omega^0\}$, where $T : \mathcal{D} \to \mathbb{R}^d$ is a mapping defined on the hold-all-domain $\mathcal{D}$ and thus an element of a natural vector space. A central requirement for the mappings representing shapes is that the respective mapping $x \mapsto T(x)$ is invertible. Otherwise, severe problems like self-intersecting discretization meshes or intersecting shape boundaries are to be expected. Furthermore, we want to concatenate shapes representing mappings, where we have to make sure that we only concatenate mappings $\mathcal{D} \to \mathcal{D}$. Thus, those T representing shapes are a subset of all possible $T : \mathcal{D} \to \mathbb{R}^d$.

We choose the framework of perturbation of identity and consider shape optimization methods, which update a shape by a vector field V, which we interpret as a mapping $V : \mathcal{D} \to \mathbb{R}^d$. Thus, we rewrite a standard shape update rule for updating a shape $\Omega^k = T^k(\Omega^0)$ by a vector field $V : \mathcal{D} \to \mathbb{R}^d$ in terms of the respective deformation mapping as

$$\Omega^{k+1} = (I + V)(\Omega^k) \Leftrightarrow T^{k+1}(\Omega^0) = (I + V) \circ T^k(\Omega^0),$$

where we observe that $I + V$ is invertible, if $\|V\|_{W^{1,\infty}} < 1$ [2], and that $I + V : \mathcal{D} \to \mathcal{D}$, if V is zero close to the boundary of $\mathcal{D}$. This is implicitly checked in each shape optimization iteration. Therefore, a standard shape optimization algorithm written as an algorithm in deformation vector fields may be written in the form of Algorithm 3.1.

At first glance, Algorithm 3.1 differs significantly from the standard steepest descent optimization algorithm in vector spaces as discussed, e.g., in [31], in view of the update rule $T^{k+1} := (I + t^k V^k) \circ T^k$. But because of the special definition of the

Algorithm 3.1 Steepest descent shape deformation optimization.

$k := 0$, initialize $T^0 = I, \varepsilon$
repeat
  determine V^k solving $b_k(V^k, Z) = -dJ(\Omega^k)[Z] \ \ \forall Z \in \mathcal{H}$
  Line search: find $t^k \approx \arg\min_t J((I + tV^k) \circ T^k(\Omega^0))$
  $T^{k+1} := (I + t^k V^k) \circ T^k$
  $k := k + 1$
until $\|V^k\| \le \varepsilon$

shape derivative, this is indeed a standard steepest descent method in deformation vector fields, as shown in the following Theorem 3.1. First we introduce the notation. We interpret the shape objective as a mapping $T \mapsto f(T) := J(T(\Omega^0))$ and rewrite the shape derivative at the domain $\Omega := T(\Omega^0)$ in terms of the perturbation vector field Z.

$$dJ(\Omega)[Z] = \frac{d}{dt}\bigg|_{t=0+} J((I+tZ)(\Omega)) = \frac{d}{dt}\bigg|_{t=0+} f((I+tZ)\circ T) = f'(T)[Z\circ T]. \tag{3.2}$$

Equation (3.2) is the central observation initiating this paper. It means that the standard shape derivative can be seen as a usual directional derivative for shape deformations, where the local directions Z are pulled back to $Z \circ T$. We would like to find a gradient related vector, which we call ∇J satisfying relation (3.1) for Z reformulated in this notation as

$$b_T(\nabla J(T), Z) = dJ(\Omega)[Z] \quad \forall Z \in \mathcal{H}\,. \tag{3.3}$$

The bilinear form $b_T(.,.)$ may be chosen differently at each point T in the deformation vector space, which is indicated by the index T. We observe that ∇J cannot be interpreted as a gradient of the function f defined on the Hilbert space of deformations $(\mathcal{H}, g)$ with scalar product g in the following way:

$$g(\nabla f(T), Z) = f'(T)Z \quad \forall Z \in \mathcal{H}\,. \tag{3.4}$$

However, there holds a revealing relation between both as formulated in Theorem 3.1.

THEOREM 3.1. *Let g be a scalar product on the linear space of mappings $\mathcal{H} := \{T \mid T : \mathcal{D} \to \mathbb{R}^d\}$ such that $(\mathcal{H}, g)$ is a Hilbert space. Let $J : \Omega \to \mathbb{R}$, where $\Omega = T(\Omega^0)$, be a shape functional and let $f : T \mapsto J(T(\Omega^0))$ be a related objective function in terms of a deformation $T \in \mathcal{H}$. Furthermore, we define ∇J by* (3.3) *and assume that all local bilinear forms $b_T(.,.)$ are related to an overall scalar product $g(.,.)$ on the space of deformations in the following natural way for any invertible $T \in \mathcal{H}$:*

$$b_T(Z_1, Z_2) := g(Z_1 \circ T, Z_2 \circ T)\,. \tag{3.5}$$

Then, there holds for any invertible $T \in \mathcal{H}$ the following relation between ∇J and the gradient of ∇f, where the gradient ∇f is defined by the scalar product g:

$$\nabla f(T) = \nabla J(T) \circ T\,. \tag{3.6}$$

Proof. We observe from (3.2), (3.3), (3.5)

$$g(\nabla J(T)\circ T, Z\circ T) = b_T(\nabla J(T), Z) = dJ(\Omega)[Z] = f'(T)[Z\circ T] \quad \forall Z \in \mathcal{H}\,.$$

Since T is assumed invertible, this is equivalent to

$$g(\nabla J(T)\circ T, Z) = f'(T)[Z] \quad \forall Z \in \mathcal{H}\,,$$

which defines ∇f according to (3.4). ∎

Therefore, algorithm (3.1) can be considered a steepest descent algorithm for the objective $f(X) := J(X(\Omega^0))$—potentially with variable metric in the case of remeshing or when using Newton's method. Thus, all standard convergence considerations apply here in the case of a finite dimensional Hilbert space $\mathcal{H}$ and natural generalizations as

known in the optimal control community apply in the case of $\mathcal{H}$ being an appropriate infinite dimensional function space. Furthermore, we note that condition (3.5) is quite natural if the scalar product $b_T(.,.)$ is defined on the finite element mesh deformed by the deformation T.

In similar fashion, we can formulate a Taylor series in the following theorem.

THEOREM 3.2. *Let the assumptions of Theorem* 3.1 *hold and assume for the vector field V defined on $\mathcal{D}$ that $I+V:\mathcal{D}\to\mathcal{D}$ is invertible and that $(I+V)\circ T\in\mathcal{H}$ for a given $T\in\mathcal{H}$. Furthermore, we assume that the function f defined in Theorem* 3.1 *is three times differentiable with bounded third derivative. Then, there holds the Taylor series*

$$J(\Omega+V(\Omega)) = J(\Omega) + b_T(\nabla J(\Omega), V) + \frac{1}{2}J''(\Omega)[V,V] + \mathcal{O}(\|V\|^3), \tag{3.7}$$

where $\mathcal{O}$ denotes the Landau symbol, the norm $\|.\|$ is derived from the scalar product $b_T(.,.)$, and $J''(\Omega)[V,W]$ is the linear second shape derivative defined by

$$J''(\Omega)[V,W] := \left.\frac{d}{ds_1}\right|_{s_1=0+} \left.\frac{d}{ds_2}\right|_{s_2=0+} J((I+s_1V+s_2W)(\Omega)). \tag{3.8}$$

Furthermore, a shape Newton method based on $J''(\Omega)[V,W]$ is equivalent to a standard Newton method for f in the following way: if V^k is the kth vector field of a shape Newton method based on $J''(\Omega)[V,W]$, such that $\Omega^{k+1}=(I+V^k)(\Omega^k)$, then $V^k\circ T^k$ is the step of a standard Newton method w.r.t. f.

Remark 3.3. Although the sum of two domains is not defined, we write in (3.7) $\Omega+V(\Omega)$ instead of the equivalent and well-defined $(I+V)(\Omega)$, in order to mimic the notation of a standard Taylor series. Here and in some instances later, we use a similar sum only in cases where it can be replaced by the equivalent application of $(I+V)$. This is done in order to improve understanding by enforcing similarity to well-known properties in linear spaces.

Remark 3.4. Under the smoothness assumptions of Theorem 3.2, the second linear shape derivative gives the same expression as the second Fréchet shape derivative as analyzed in [32]. Thus, the Taylor expansion itself does not come as a surprise. In contrast to [32], we are analyzing large deformations as a consequence of several steps of a shape optimization algorithm. Thus, the novelty of Theorem 3.2 lies in the observation that shape Newton iterations seemingly defined differently at each iteration can be embedded into an overall Newton iteration in the space of deformations.

Proof. The assumptions guarantee the Taylor series for f, which is now rephrased. There holds

$$\begin{aligned}
J(\Omega+V(\Omega)) &= f(T+V\circ T)\\
&= f(T) + f'(T)[V\circ T] + \frac{1}{2}f''(T)[V\circ T, V\circ T] + \mathcal{O}(\|V\circ T\|^3)\\
&= f(T) + g(\nabla f(T), V\circ T)\\
&\quad + \frac{1}{2}\left.\frac{d}{ds_1}\right|_{s_1=0+}\left.\frac{d}{ds_2}\right|_{s_2=0+} f(T+s_1V\circ T+s_2V\circ T) + \mathcal{O}(\|V\circ T\|^3)\\
&= J(\Omega) + b_T(\nabla J(\Omega), V) + \frac{1}{2}J''(\Omega)[V,V] + \mathcal{O}(\|V\circ T\|^3).
\end{aligned}$$

The norm in the remainder term is derived from the scalar product g. If we translate it to $b_T(.,.)$, we confirm the assertion.

Now, we formulate a shape Newton method

$$\Omega^{k+1} = \Omega^k + V^k(\Omega^k), \text{ where } V^k \text{ is defined by}$$
$$J''(\Omega^k)[V^k, W] = -dJ(\Omega^k)[W] \quad \forall W \in \mathcal{H},$$

which is thus equivalent to the formulation in deformation vector space

$$T^{k+1} = T^k + V^k \circ T^k, \text{ where } V^k \text{ is defined by}$$
$$f''(T^k)[V^k \circ T^k, W] = -f'(T^k)[W] \quad \forall W \in \mathcal{H}$$

with the definition $\Omega^{k+1} = T^{k+1}(\Omega)$ and $\Omega^k = T^k(\Omega)$. ∎

The Taylor series allows the construction of linear second order optimization methods (see below) and the investigation of well-posedness of shape optimization problems, although well-posedness considerations are carried out on the shape boundary in a simpler way, either in terms of local parametrization as in [14] or in terms of shape manifolds as in [37]. It should be noted that the linear second shape derivative $J''(\Omega)[V, W]$ defined in Theorem 3.2, which is needed for the Taylor series, is symmetric and differs from the classical shape Hessian, which we recall here as

$$d^2J(\Omega)[V, W] = \frac{d}{ds_2}\bigg|_{2=0+} \frac{d}{ds_1}\bigg|_{s_1=0+} J((I + s_1V) \circ (I + s_2W)(\Omega)).$$

The difference contains several nonsymmetric derivative terms of $V \circ W$ arising in the classical shape Hessian. It is shown in [9, p. 504] that the classical shape Hessian is symmetric if $\mathrm{D}\,VW = \mathrm{D}\,WV$. The usage of the linear second shape derivative above makes this somewhat artificial assumption no longer necessary.

Unfortunately, J'' and thus f'' have a nontrivial and even huge null-space, which cannot simply be ignored. If this would not be the case, then standard arguments would lead to locally quadratic convergence of the shape Newton method. In order to mitigate this problem, we use the Moore–Penrose pseudoinverse instead, which allows for locally quadratic convergence again. Since the convergence theory is more easily formulated for operators than for bilinear forms, we use again the Riesz representation theorem in the Hilbert space $(\mathcal{H}, g)$ and define the gradient as in (3.6) and with that also the Hessian operator $H(T)$ is defined by the operator representation of $f''(T)$, i.e.,

$$g(H(T)V, W) := f''(T)[V, W] \quad \forall V, W \in \mathcal{H}.$$

In this notation, the following iteration is performed:

$$T^{k+1} = T^k + V^k \circ T^k, \text{ where } V^k = -H(T^k)^+ \nabla f(T^k). \tag{3.9}$$

Here $H(T^k)^+$ denotes the Moore–Penrose pseudoinverse of $H(T^k)$. Specific aspects of the implementation are discussed in Theorem 3.5 below. Iteration (3.9) can be shown to be locally quadratically convergent by applying well-known results from linear spaces as in [12].

The Moore–Penrose pseudoinverse based on the SVD with the standard Euclidean scalar product is computationally inefficient. Therefore, we reformulate the Moore–Penrose pseudoinverse as a least squares problem in the scalar product related to the Hilbert space we are working in and investigate a perturbation approach. We are considering the linear equation $HV = d$, where $H = H(T^k)$ is the Hessian operator and $d = \nabla f(T^k)$ is the gradient at step k of iteration (3.9), i.e., a Newton method based on the pseudoinverse.

THEOREM 3.5. *Let $(\mathcal{H}, g)$ be a Hilbert space with inner product g. We assume that the linear operator H defined on $\mathcal{H}$ has closed range and is not necessarily invertible. When solving the equation $HV = d$ with $d \in \mathcal{R}(H)$, we obtain $\hat{V} := H^+ d$ as the minimum norm solution, where H^+ is the Moore–Penrose pseudoinverse operator. Then, the vector $\hat{V}$ is also the unique solution of the optimization problem*

$$\min_V \; g(V,V) \tag{3.10}$$

$$s.t. \quad HV = d. \tag{3.11}$$

Furthermore, if H is self-adjoint in the scalar product g and positive semidefinite, then the vector $\hat{V}$ can be computed as the limit of the solutions V_ε of the following family of linear-quadratic problems parameterized by $\varepsilon > 0$. For

$$V_\varepsilon := \arg\min_V \frac{1}{2} g(HV, V) - g(d, V) + \frac{\varepsilon}{2} g(V, V) \tag{3.12}$$

there holds $\lim_{\varepsilon \to 0} V_\varepsilon = \hat{V}$.

Proof. Instead of the frequently used definition of the pseudoinverse by an SVD, we use the equivalent variational definition from [19, Definition (V), p. 45], which defines it as the minimum norm solution of (3.11) in least squares reformulation. Thus any solution of the equation $HV = d$ can be written as $V = H^+ d + V_N = \hat{V} + V_N$, where $V_N \in \mathcal{N}(H)$. We observe that $\hat{V} \in \mathcal{R}(H^+) = \mathcal{R}(H^*)$ [19, Theorem 2.1.2], where H^* is the adjoint operator. Therefore $\hat{V} \perp V_N$ for any $V_N \in \mathcal{N}(A)$ [19, Theorem 1.2.1], where $\perp$ means orthogonality in the g scalar product. Using this in the objective (3.10) gives the assertion of the first part.

The necessary conditions of problem (3.12) can be written as

$$\begin{aligned} & g(HV_\varepsilon, \eta) - g(d, \eta) + \varepsilon g(V_\varepsilon, \eta) = 0 \quad \forall \eta \in \mathcal{H} \\ \Leftrightarrow \quad & HV_\varepsilon - d + \varepsilon V_\varepsilon = 0. \end{aligned}$$

We use the Ansatz $V_\varepsilon =: \hat{V} + Y_\varepsilon$ for some $Y_\varepsilon \in \mathcal{H}$. Since $H\hat{V} - d = 0$, we conclude

$$(H + \varepsilon I) Y_\varepsilon = -\varepsilon \hat{V}. \tag{3.13}$$

Since $\hat{V} \in \mathcal{R}(H)$, because H is self-adjoint, also $Y_\varepsilon \in \mathcal{R}(H)$ as a first consequence of (3.13), for all $\varepsilon > 0$. On the other hand, we observe in the same equation $HY_\varepsilon \to 0$ for $\varepsilon \to 0$ and thus $\lim_{\varepsilon \to 0} Y_\varepsilon \in \mathcal{N}(H)$. Therefore

$$V_\varepsilon - \hat{V} = Y_\varepsilon \to 0 \text{ for } \varepsilon \to 0.$$

∎

In the context of a shape Newton method H is the Hessian operator related to the linearized second shape derivative and d is the gradient of the shape derivative.

COROLLARY 3.6. *Due to the definition of the Hessian and the gradient operator,* (3.12) *is obviously equivalent to the formulation*

$$\min_V \frac{1}{2} f''(T^k)[V \circ T^k, V \circ T^k] - f'(T^k)[V \circ T^k] + \frac{\varepsilon}{2} g(V \circ T^k, V \circ T^k), \tag{3.14}$$

which again, due to the definition of f, can be equivalently rephrased in terms of the shape functional as

$$\min_V \frac{1}{2} J''(\Omega^k)[V, V] - dJ(\Omega^k)[V] + \varepsilon b_{\Omega^k}(V, V), \tag{3.15}$$

which has to be solved in each shape Newton iteration.

If ε is reduced during the Newton iterations in the fashion $\varepsilon_k = \mathcal{O}(\|V^k\|)$, then locally quadratic convergence of the resulting method can be expected from respective investigations in full Newton methods [11]. However, due to discretization effects, the condition in Theorem 3.5 that the gradient has to lie in the range of the Hessian may not be fulfilled exactly in the discretized equations. Therefore ε may not be chosen arbitrarily small in practice. Formulations (3.12), (3.14), (3.15) are related to Tikhonov regularization. Thus, a similar result can be found in [19, Corollary 2.3.8] for the assumption that H^*H is invertible, which does not hold here in general.

A different approach for the computation of the operator-vector product H^+d can be performed by the usage of Krylov subspace methods; see, e.g., [21, 8, 33]. However, the relevant publications have to be rephrased to take into account a general scalar product g rather than the Euclidean scalar product.

4. Material derivatives as test vectors. So far, we have discussed derivatives of mappings from shapes to an objective criterion. Often, this mapping involves system equations in the form of PDEs, where it is reasonable to formulate the PDE explicitly as a constraint. Because of the intricacies of the shape calculus, it seems difficult to derive necessary optimality conditions for system model based shape optimization problems. Here, we introduce a simple step-by-step procedure for the derivation of the weak form of necessary conditions. We start out with the general necessary conditions of optimality for model based optimization problems in the form

$$\begin{aligned} &\min J(u,q) \\ &\text{s.t. } (c(u,q),\lambda) = 0 \ \ \forall \lambda \in \Lambda. \end{aligned}$$

There, we build the Lagrangian

$$\mathscr{L}(u,q,\lambda) := J(u,q) + (c(u,q),\lambda)$$

and observe as necessary conditions of optimality

$$\frac{\partial}{\partial u}\mathscr{L}(u,q,\lambda)[\tilde{u}] = 0 \ \ \forall \tilde{u} \in U, \tag{4.1}$$

$$\frac{\partial}{\partial q}\mathscr{L}(u,q,\lambda)[\tilde{q}] = 0 \ \ \forall \tilde{q} \in Q, \tag{4.2}$$

$$\frac{\partial}{\partial \lambda}\mathscr{L}(u,q,\lambda)[\tilde{\lambda}] = 0 \ \ \forall \tilde{\lambda} \in \Lambda, \tag{4.3}$$

where often $U = \Lambda$. Conditions (4.1)–(4.3) can be equivalently and jointly written as

$$\frac{\partial}{\partial u}\mathscr{L}(u,q,\lambda)[\tilde{u}] + \frac{\partial}{\partial q}\mathscr{L}(u,q,\lambda)[\tilde{q}] + \frac{\partial}{\partial \lambda}\mathscr{L}(u,q,\lambda)[\tilde{\lambda}] = 0 \ \ \forall (\tilde{u},\tilde{q},\tilde{\lambda}) \in U \times Q \times \Lambda \tag{4.4}$$

since each individual equation from (4.1)–(4.3) follows from this equation by setting two out of the three vectors $\tilde{u}, \tilde{q}, \tilde{\lambda}$ to zero. We arrive at the same expression on the left-hand side by explicitly denoting at the solution of the optimization problem that the state and adjoint depend on the optimization variable, i.e., $u = u(q), \lambda = \lambda(q)$, and calculating the total derivative of the Lagrangian with respect to q as

$$\begin{aligned} &d\mathscr{L}(u(q),q,\lambda(q))[\tilde{q}] \\ &= \frac{\partial}{\partial u}\mathscr{L}(u,q,\lambda)[du[\tilde{q}]] + \frac{\partial}{\partial q}\mathscr{L}(u,q,\lambda)[\tilde{q}] + \frac{\partial}{\partial \lambda}\mathscr{L}(u,q,\lambda)[d\lambda[\tilde{q}]]. \end{aligned}$$

In this way, $d\mathscr{L}[\tilde{u}]$ depends additionally on the vectors $du[\tilde{q}], d\lambda[\tilde{q}]$, which thus can be seen as somewhat fancy notation for the vectors $\tilde{u}, \tilde{\lambda}$ in (4.4).

For standard model based opimization, this point of view is redundant. However, in the context of shape calculus, this approach gives a convenient guideline for deriving necessary conditions in the following steps:

1. build a Lagrangian $\mathscr{L}(u, \Omega, \lambda)$ of the PDE constrained shape optimization problem based on state variable u and adjoint λ,
2. derive an expression for its shape derivative, where material derivatives $\mathrm{d_M}\, u[V], \mathrm{d_M}\, \lambda[V]$ explicitly appear, and denote this expression as $d\mathscr{L}(u, \Omega, \lambda)[\mathrm{d_M}\, u[V], V, \mathrm{d_M}\, \lambda[V]]$,
3. state necessary conditions in weak form as

$$d\mathscr{L}(u, \Omega, \lambda)[\mathrm{d_M}\, u[V], V, \mathrm{d_M}\, \lambda[V]] = 0 \ \ \forall (\mathrm{d_M}\, u[V], V, \mathrm{d_M}\, \lambda[V]) \in U \times H \times V,$$

where U is the space for the state variables, H is the Hilbert space for the admissible deformation vector fields of the shape, and V is the space for the adjoint variables in the respective shape optimization problem.

This process is carried out and exemplified in detail in the next section and in Appendix A.

Remark 4.1. Of course, in principle, also expressions for $d\mathscr{L}$ can be derived, where local shape derivatives instead of material derivatives appear. However, as noted in [6], only material derivatives can be considered from the same vector space as the state or adjoint variable. Furthermore, the existence of the material derivative of the adjoint might be in question. In this case, the derivation following steps 1–3 above can be considered only formal. Nevertheless, the resulting necessary conditions are in line with a derivation avoiding material derivatives at all as discussed in [42].

5. Model problem. Shape optimization methods based on second order information are discussed in [1, 13], where electrical impedance tomography is treated. [1] investigates differentiability and well-posedness based on the shape Hessian. In [13] star-shaped domains are considered and respectively parameterized. The resulting Hessian is approximated by a Fourier series and Newton variants based on that approximation are tested. Inspired by these investgations, we use a related simpler model problem, in order to illustrate the effects discussed in the previous sections. We seek to reconstruct the shape of an inclusion Ω_0 (see Figure 1) within the surrounding domain Ω_1 with $[0,1]^2 =: \Omega = \Omega_0 \cup \Omega_1$ disjointly. The actual reconstruction differs from electrical impedance tomography, since we assume measurements in the domain rather than at the boundary. It is given by the problem

$$\begin{aligned} &\min_{(u,\Omega)} \quad \frac{1}{2}\int_\Omega (u-z)^2 \,\mathrm{d}\, x + \alpha R(\Omega) \\ &\text{s.t. find } u \in u_0 + H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega) \text{ such that} \\ &\int_\Omega \mu \left\langle \nabla u, \nabla v \right\rangle_2 - fv \ \,\mathrm{d}\, x = 0 \quad \forall v \in H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega), \end{aligned} \tag{5.1}$$

where $R(\Omega)$ stands for a regularization term, i.e., the perimeter of $\partial\Omega_0$, i.e.,

$$R(\Omega) := \int_{\partial\Omega_0} 1 \,\mathrm{d}\, s.$$

In the numerical examples below, we choose $\alpha = 10^{-6}$. The affine Sobolev space $u_0 + H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega)$, where $u_0(x_1, x_2) := x_2$ and $H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega) = \{\varphi \in H^1(\Omega) : \varphi|_{\Gamma_0 \cup \Gamma_2} = 0\}$, incorporates the inhomogeneous Dirichlet boundary conditions

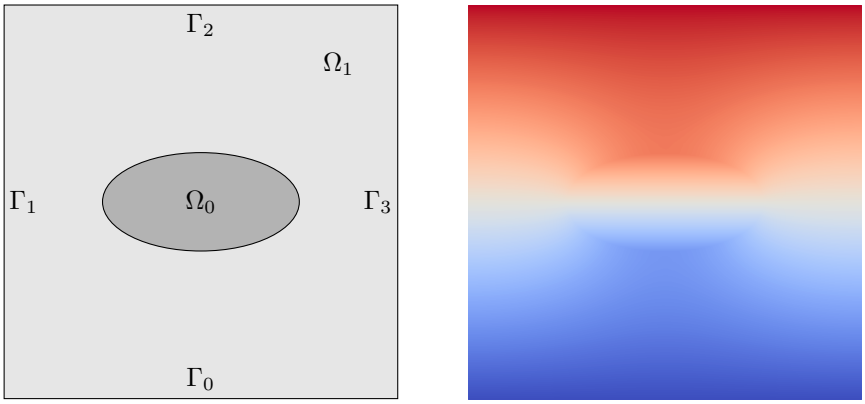

FIG. 1. *The interface identification problem. Domain Ω_0 to be reconstructed on the left, corresponding target state z on the right. (Color available online.)*

$$u = 0 \text{ on } \Gamma_0,$$
$$u = 1 \text{ on } \Gamma_2,$$

while the "do-nothing-condition" naturally creates a homogeneous Neumann condition on the remaining boundaries. These boundary conditions can be interpreted as putting an electric potential between the upper and lower sides of the domain. The actual electric field lines would then be given by ∇u. As such, the Neumann boundary conditions can be interpreted as no electric field lines entering or leaving the domain. The conductivity μ directly depends on the subdomains Ω_0 and Ω_1 via

$$\mu(x) = \begin{cases} 10^{-6}, & x \in \Omega_0, \\ 1, & x \in \Omega_1. \end{cases} \tag{5.2}$$

We generate the desired potential z by solving the problem beforehand for an elliptic inclusion Ω_0 as shown in Figure 1.

Knowledge of the actual target domain layout is then discarded and, starting from a circle, we move the interior boundary $\partial\Omega_0$ with the intention to recover z from the newly calculated u.

The Lagrangian of this problem for $u \in u_0 + H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega)$ and $\lambda \in H^1_{(\Gamma_0 \cup \Gamma_2)}(\Omega)$ is

$$\mathscr{L}(u, \Omega, \lambda) = \frac{1}{2} \int_\Omega (u - z)^2 \,\mathrm{d}\, x + \alpha \int_{\partial\Omega_0} 1 \,\mathrm{d}\, s + \mu \int_\Omega \nabla u^\top \nabla \lambda - f\lambda \,\mathrm{d}\, x.$$

We use the principles presented in section 4 in order to derive first order shape derivatives. Step 1 is already accomplished by building the Lagrangian above. This is now differentiated with respect to the shape (more details can be found in Appendix A)

$$\begin{aligned}
&\mathrm{d}\,\mathscr{L}(u, \Omega, \lambda)[\mathrm{d_M}\, u[V], V, \mathrm{d_M}\, \lambda[V]] \\
&\quad = \int_\Omega \operatorname{div}(V) \left(\frac{1}{2}(u - z)^2 + \mu \langle \nabla u, \nabla \lambda \rangle - \lambda f \right) \,\mathrm{d}\, x \\
&\quad + \int_\Omega \mathrm{d_M} \left(\frac{1}{2}(u - z)^2 + \mu \langle \nabla u, \nabla \lambda \rangle - \lambda f \right)[V] \,\mathrm{d}\, x + \alpha \int_{\partial\Omega_0} \operatorname{div}_\Gamma V \,\mathrm{d}\, s \\
&\quad = \int_\Omega \operatorname{div}(V) \left(\frac{1}{2}(u - z)^2 + \mu \langle \nabla u, \nabla \lambda \rangle - \lambda f \right) \,\mathrm{d}\, x
\end{aligned}$$

$$+\int_\Omega (u-z)(\mathrm{d}_\mathrm{M} u[V]-\mathrm{d}_\mathrm{M} z[V])+\mu \mathrm{d}_\mathrm{M}\langle\nabla u,\nabla\lambda\rangle[V]+\langle\nabla u,\nabla\lambda\rangle \mathrm{d}_\mathrm{M}\mu[V]\ \mathrm{d}x$$
$$+\int_\Omega -f\mathrm{d}_\mathrm{M}\lambda[V]-\lambda\mathrm{d}_\mathrm{M} f[V]\ \mathrm{d}x+\alpha\int_{\partial\Omega_0}\mathrm{div}_\Gamma V\ \mathrm{d}s.$$

Now, we use the well-known identity

$$\mathrm{d}_\mathrm{M}\langle\nabla u,\nabla\lambda\rangle[V]=\langle\nabla \mathrm{d}_\mathrm{M} u[V],\nabla\lambda\rangle+\langle\nabla u,\nabla \mathrm{d}_\mathrm{M}\lambda[V]\rangle-\left\langle\nabla u,(\mathrm{D}V+\mathrm{D}V^T)\nabla\lambda\right\rangle.$$

Thus, the Lagrangian can be rephrased as

(5.3)
$$\begin{aligned}&\mathrm{d}\mathscr{L}(u,\Omega,\lambda)[\mathrm{d}_\mathrm{M} u[V],V,\mathrm{d}_\mathrm{M}\lambda[V]]\\&\quad=\int_\Omega \mathrm{div}(V)\left(\frac{1}{2}(u-z)^2+\mu\langle\nabla u,\nabla\lambda\rangle-\lambda f\right)-(u-z)\mathrm{d}_\mathrm{M} z[V]-\lambda\mathrm{d}_\mathrm{M} f[V]\\&\qquad-\mu\left\langle\nabla u,(\mathrm{D}V+\mathrm{D}V^T)\nabla\lambda\right\rangle+\langle\nabla u,\nabla\lambda\rangle\mathrm{d}_\mathrm{M}\mu[V]\ \mathrm{d}x\\&\qquad+\int_\Omega (u-z)\mathrm{d}_\mathrm{M} u[V]+\mu\langle\nabla \mathrm{d}_\mathrm{M} u[V],\nabla\lambda\rangle+\mu\langle\nabla u,\nabla \mathrm{d}_\mathrm{M}\lambda[V]\rangle-f\mathrm{d}_\mathrm{M}\lambda[V]\ \mathrm{d}x\\&\qquad+\alpha\int_{\partial\Omega_0}\mathrm{div}_\Gamma V\ \mathrm{d}s.\end{aligned}$$

The condition $\mathrm{d}\mathscr{L}(u,\Omega,\lambda)[\mathrm{d}_\mathrm{M} u[V],V,\mathrm{d}_\mathrm{M}\lambda[V]]=0$ for all $\mathrm{d}_\mathrm{M}\lambda[V]\in H^1_{(\Gamma_0\cup\Gamma_2)}(\Omega)$ gives the state equation back. The condition $\mathrm{d}_\mathrm{M}\mathscr{L}(u,\Omega,\lambda)[\mathrm{d}_\mathrm{M} u[V],V,\mathrm{d}_\mathrm{M}\lambda[V]]=0$ for all $\mathrm{d}_\mathrm{M} u[V]\in H^1_{(\Gamma_0\cup\Gamma_2)}(\Omega)$ results in the adjoint equation

$$\int_\Omega \mu\langle\nabla \mathrm{d}_\mathrm{M} u[V],\nabla\lambda\rangle+(u-z)\mathrm{d}_\mathrm{M} u[V]\ \mathrm{d}x=0\quad\forall \mathrm{d}_\mathrm{M} u[V]\in H^1_{(\Gamma_0\cup\Gamma_2)}(\Omega). \tag{5.4}$$

By using the state and adjoint equations, we obtain the simplified shape derivative of the Lagrangian,

(5.5)
$$\begin{aligned}&\mathrm{d}\mathscr{L}(u,\Omega,\lambda)[V]\\&\quad=\int_\Omega \mathrm{div}(V)\left(\frac{1}{2}(u-z)^2+\mu\langle\nabla u,\nabla\lambda\rangle-\lambda f\right)-(u-z)\mathrm{d}_\mathrm{M} z[V]-\lambda\mathrm{d}_\mathrm{M} f[V]\\&\qquad-\mu\left\langle\nabla u,(\mathrm{D}V+\mathrm{D}V^T)\nabla\lambda\right\rangle+\langle\nabla u,\nabla\lambda\rangle\mathrm{d}_\mathrm{M}\mu[V]\ \mathrm{d}x+\alpha\int_{\partial\Omega_0}\mathrm{div}_\Gamma V\ \mathrm{d}s.\end{aligned}$$

In the particular case of the numerical example, there is $f\equiv 0$, z is a fixed field, and μ deforms with the shape, and thus $\mathrm{d}_\mathrm{M} f[V]=0,\mathrm{d}_\mathrm{M}\mu[V]=0,\mathrm{d}_\mathrm{M} z[V]=\nabla z^\top V$, which simplifies the shape derivative even more to

(5.6)
$$\begin{aligned}&\mathrm{d}\mathscr{L}(u,\Omega,\lambda)[V]\\&\quad=\int_\Omega \mathrm{div}(V)\left(\frac{1}{2}(u-z)^2+\mu\langle\nabla u,\nabla\lambda\rangle-\lambda f\right)-(u-z)\nabla z^\top V\\&\qquad-\mu\left\langle\nabla u,(\mathrm{D}V+\mathrm{D}V^T)\nabla\lambda\right\rangle\ \mathrm{d}x+\alpha\int_{\partial\Omega_0}\mathrm{div}_\Gamma V\ \mathrm{d}s.\end{aligned}$$

Here, we assume that u and λ satisfy the primal and adjoint equations.

The linear second shape derivative $J''(\Omega)[V,W]$ is derived in Appendix A.

6. Numerical results. The Taylor series expansion shown in Theorem 3.2 gives the potential of quadratic convergence for shape optimization algorithms using the

linear second shape derivative introduced there, since the discussion before this theorem shows that standard shape optimization can be viewed from a linear vector space perspective. The performance obstacle is, however, the lack of positive definiteness of the linear second shape derivative. Without regularization, the publication [1] shows lack of definiteness of the shape Hessian for the boundary formulation in electrical impedance tomography. This aspect is acerbated here in the volume formulation used in this paper. Shape gradient algorithms based on the volume formulation can sometimes beat solution strategies based on the boundary formulation as shown in [46]. The boundary formulation here can also be employed, although it holds less generally in the sense that it requires more smoothness and is less accurate when finite element discretization is used as shown in [23]. If boundary correction is used, however, boundary formulation after finite element discretization can have similar numerical accuracy as the volume formulation [18].

Corollary 3.6 specifies a convenient strategy to circumvent the lack of definiteness of the linear second shape derivative at the cost of loosing quadratic convergence, but getting almost arbitrarily good linear convergence. The aim of this section is to illustrate this effect in numerical computations for the model problem introduced in section 5. Here, we first give more details on the algorithmic realization and afterward illustrate the interplay of Theorem 3.2 and Corollary 3.6.

The variational Newton method aims at solving the stationarity condition for the Lagrangian with respect to all variables. Thus, it is a method iterating over all variables simultaneously in the form

$$(u,\Omega,\lambda)^{k+1} = (u,\Omega,\lambda)^k + (\hat{u},\hat{V}(\Omega^k),\hat{\lambda})\,,$$

where $\Omega^k + \hat{V}(\Omega^k)$ has to be read as $(I+\hat{V})(\Omega^k)$ and $(\hat{u},\hat{V},\hat{\lambda})$ solve the variational problem for the Lagrangian $\mathscr{L}$ defined in the previous section, discretized by continuous finite elements

$$\begin{aligned}(6.1)\qquad \mathscr{L}''(u^k,\Omega^k,\lambda^k)[\tilde{u},\tilde{V},\tilde{\lambda}][\hat{u},\hat{V},\hat{\lambda}] &= -\,\mathrm{d}\mathscr{L}(u^k,\Omega^k,\lambda^k)[\tilde{u},\tilde{V},\tilde{\lambda}]\\ &\qquad \forall(\tilde{u},\tilde{V},\tilde{\lambda}) \in \mathrm{CG}^1_{r_1}\times\mathrm{CG}^2_{r_2}\times\mathrm{CG}^1_{r_1},\end{aligned}$$

where $\mathrm{CG}^m_n := \{p\in C^0(\Omega_h,\mathbb{R}^m) : p_i|_T \in P_n(T),\, i=1,\dots,d, \forall T\in\Omega_h\}$. Here, $P_n(T)$ is the set of polynomials of order n over the simplex T of a nonoverlapping mesh Ω_h. Equipping this space with the classical nodal basis, we arrive at the classical continuous Lagrange finite elements. Here, $\mathscr{L}''$ means the linear second order derivatives as in Theorem 3.2, which is only of importance for the shape part, of course. In this way nonsymmetric terms of type $\tilde{V}\circ\hat{V}$ do not arise. As discussed above, this is just a particular formulation of a Newton method iterating over the state/adjoint variables and the deformation vector field of the shape. Thus, classical Newton convergence theory applies in linear spaces.

Equation (6.1) is formulated for a particular and typical choice of ansatz and test spaces, which can be, of course, adapted to a specific application. Any solution algorithm has to cope with the fact that the shape Hessian has a huge kernel due to the Hadamard structure theorem (cf., e.g., [9, Theorem 3.6]). As already discussed, the simplest approach is to add a Tikhonov type regularization term involving a coercive bilinear form b_{Ω^k} such that the regularized variant

$$\begin{aligned}(6.2)\qquad \mathscr{L}''(u^k,\Omega^k,\lambda^k)[\tilde{u},\tilde{V},\tilde{\lambda}][\hat{u},\hat{V},\hat{\lambda}] + \varepsilon b_{\Omega^k}(\tilde{V},\hat{V}) &= -\,\mathrm{d}\mathscr{L}(u^k,\Omega^k,\lambda^k)[\tilde{u},\tilde{V},\tilde{\lambda}]\\ &\qquad \forall(\tilde{u},\tilde{V},\tilde{\lambda}) \in \mathrm{CG}^1_{r_1}\times\mathrm{CG}^2_{r_2}\times\mathrm{CG}^1_{r_1}\end{aligned}$$

possesses a unique solution, which converges to the Moore–Penrose pseudoinverse solution for $\varepsilon \to 0$; cf. Remark 3.6.

6.1. Numerical results with fixed stabilization. With obvious abbreviations, in particular $\mathscr{L}_\Omega$ and $\mathscr{L}_{\Omega\Omega}$ for first and second linear shape derivatives, the variational KKT system (6.2) with stabilization can be written in matrix form as

$$\begin{bmatrix} \mathscr{L}_{uu} & \mathscr{L}_{u\Omega} & \mathscr{L}_{u\lambda} \\ \mathscr{L}_{\Omega u} & \mathscr{L}_{\Omega\Omega} + \varepsilon b_\Omega & \mathscr{L}_{\Omega\lambda} \\ \mathscr{L}_{\lambda u} & \mathscr{L}_{\lambda\Omega} & 0 \end{bmatrix} \begin{pmatrix} \hat{u} \\ \hat{V} \\ \hat{\lambda} \end{pmatrix} = - \begin{pmatrix} \mathscr{L}_u \\ \mathscr{L}_\Omega \\ \mathscr{L}_\lambda \end{pmatrix}.$$

The regularization term B is chosen as the matrix representation of the form

$$b_\Omega(W, V) = \int_\Omega \langle W, V \rangle_2 + \langle \nabla W, \nabla V \rangle_F \, \mathrm{d}\, x,$$

where we set $\varepsilon := 10^{-1}$ throughout the entire optimization.

The initial guess, a circle for the geometry and zeros for state and adjoint, is outside the convergence radius of the Newton method. To this end, we damp the shape update of the first five iterations, but not the update for state and adjoint, i.e., we set

$$(u^{k+1}, V^{k+1}, \lambda^{k+1}) = (u^k + \hat{u}, V^k + \delta \hat{V}, \lambda^k + \hat{\lambda})$$

with $\delta = 5 \cdot 10^{-1}$. The resulting convergence behavior for four meshes with increasing resolution is shown in Figure 2. More details on dampening the design update only can, for example, also be found in [17, 38]. One can observe that although the fairly large stabilization of $\varepsilon = 10^{-1}$ makes the Hessian for the far-off initial guess indeed positive definite, it unfortunately also makes the method behave too much like a gradient descent in the long run. To this end, we have also studied an approach, where ε is driven to zero as the method convergence as suggested in [11].

6.2. Results with adaptive stabilization. As a second test, we make the stabilization term εb_Ω vanish as the residual decreases. To this end, we now couple ε to the residual in the kth iteration via

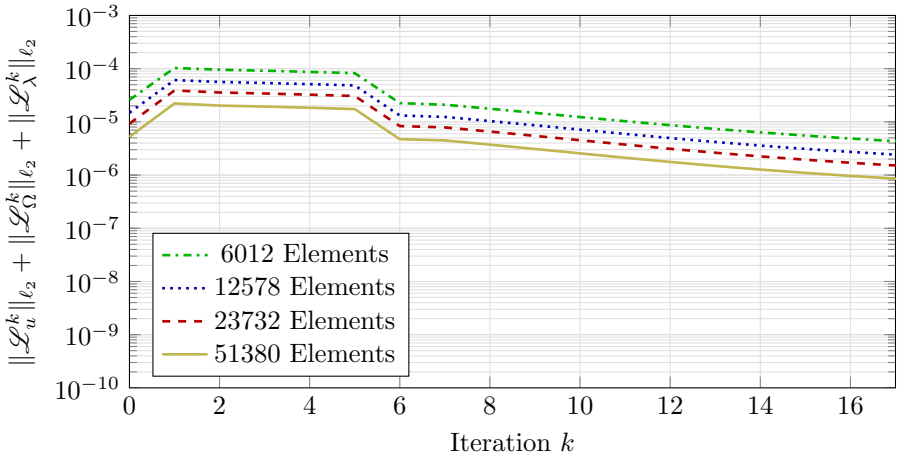

FIG. 2. *Effects of the Newton-type algorithm for constant stabilization* $\varepsilon = 10^{-1}$. *The update for the shape, but not for state and adjoint, is damped by* $5 \cdot 10^{-1}$ *until iteration* 5.

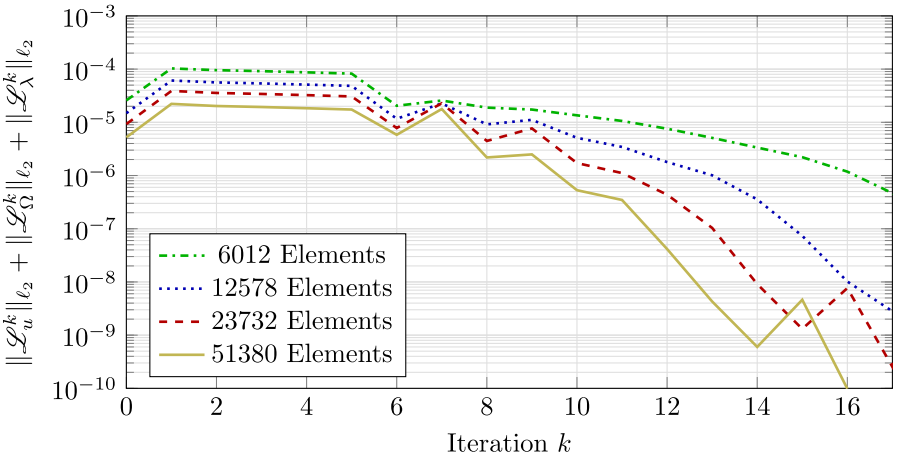

FIG. 3. *Effects of the Newton-type algorithm for adaptive stabilization. Shape updates are damped and stabilization is fixed until iteration* 5.

$$\varepsilon^k := 4 \cdot 10^3 \cdot \left(\|\mathscr{L}_u^k\|_{\ell_2} + \|\mathscr{L}_\Omega^k\|_{\ell_2} + \|\mathscr{L}_\lambda^k\|_{\ell_2} \right).$$

We again leave the start-up phase untouched with a damped shape update of $\delta = 5 \cdot 10^{-1}$ and a constant stabilization of $\varepsilon = 10^{-1}$. After iteration 5, the update is undamped and ε is adaptively coupled to the residual as described above. The convergence behavior is shown in Figure 3.

After iteration 5, we observe superlinear convergence for the two finest meshes. The two coarser meshes result in an oscillatory behavior beyond iteration 17, because the target domain also uses the 51,380 elements resolution and is unobtainable for the coarser meshes. Hence, the regularization α would need to be chosen larger than 10^{-6} should one want to pursue the further use of such coarse meshes.

All computations were performed on an unstructured mesh, where the shape optimization algorithms have been implemented by using the FE toolbox FEniCS [3] and the code used to compute the numerical results is publicly available at https://bitbucket.org/Epoxid/laplace_tracking

7. Conclusions. This paper analyzes standard descending shape optimization algorithms from the point of view of iterating over deformations rather than geometries. It turns out that standard choices for scalar products in Hilbert spaces of deformations show indeed equivalence of shape optimization algorithms to optimization in a Hilbert space of deformations. Based on that, a linear variant of the second shape derivative is employed, on which one can base a Taylor expansion in linear spaces as well as a standard Newton iteration. The central obstacle of this point of view is the significant rank deficiency of the linear shape Hessian. As a remedy, Newton-type methods based on pseudoinverses are presented, as well as a convenient way to substitute the otherwise numerically challenging pseudoinverse.

Finally, the question remains of whether we should replace standard shape optimization algorithms of the form of Algorithm 3.1 by a rather more straightforward iteration in the linear Hilbert space of deformations. The catch is that all intermediate and also the final domain deformation have to be invertible. This property can be much more easily be checked in each iteration of Algorithm 3.1 separately rather than employing a standard formulation of a descent algorithm in vector spaces. Thus, we

still recommend algorithms of the Algorithm 3.1 type, but to keep in mind that the linear point of view provides a very convenient tool for the analysis of these algorithms.

Appendix A. Second derivatives for the interface problem.. Here, the full Newton KKT system is derived in variational form. As discussed in Remark 4.1, using material derivatives will be conformal with classical continuous Lagrange finite elements. From section 3 it is clear that we need the linear second shape derivative $\mathscr{L}''$. This is, however, derived in the following manner. We investigate the standard second shape derivative $d^2\mathscr{L}$ and exclude the nonsymmetric terms on the way. Indeed, in order to derive the KKT system, the steps for the first shape derivative can very much be repeated to derive higher order shape derivatives. The volume formulation thus makes considering the Hessian almost as elegant as deriving the gradient.

Recall that in a general setting, the volume form of the shape derivative for a volume objective is given by [40, sections 2.30 and 2.32]

$$\mathrm{d}\, J(\Omega)[V] = \int_\Omega g \operatorname{div} V + \mathrm{d}_{\mathrm{M}}\, g[V] \;\mathrm{d}\, x.$$

Applying the same transformation again, one arrives at

$$\mathrm{d}^2\, J[V,W] = \int_\Omega (g \operatorname{div} V + \mathrm{d}_{\mathrm{M}}\, g[V]) \operatorname{div} W + \mathrm{d}_{\mathrm{M}}\, (g \operatorname{div} V + \mathrm{d}_{\mathrm{M}}\, g[V])[W] \;\mathrm{d}\, x. \tag{A.1}$$

Using the rule $\mathrm{d}_{\mathrm{M}}\, (\mathrm{D}\, u)[V] = \mathrm{D}(\mathrm{d}_{\mathrm{M}}\, u[V]) - \mathrm{D}\, u\, \mathrm{D}\, V$ for a sufficiently smooth function u, one can see that

$$\begin{aligned}\mathrm{d}_{\mathrm{M}}\, (\operatorname{div} V)[W] &= \mathrm{d}_{\mathrm{M}}\, (\operatorname{tr} \mathrm{D}\, V)[W] = \operatorname{tr} (\mathrm{d}_{\mathrm{M}}\, (\mathrm{D}\, V)[W]) = \operatorname{tr} (\mathrm{D}\, \mathrm{d}_{\mathrm{M}}\, V[W] - \mathrm{D}\, V\, \mathrm{D}\, W) \\ &= \operatorname{div} (\mathrm{d}_{\mathrm{M}}\, V[W]) - \operatorname{tr}(\mathrm{D}\, V\, \mathrm{D}\, W).\end{aligned}$$

Within the finite element context, one can consider the mesh motions V and W to be given as vector-Lagrange functions in $\mathrm{CG}_1^3(\mathcal{D})$. Hence, they are automatically transported with the mesh and one immediately arrives at $\mathrm{d}_{\mathrm{M}}\, V[W] = 0$. This simplifies the above within the finite element context to

$$\mathrm{d}_{\mathrm{M}}\, (\operatorname{div} V)[W] = -\operatorname{tr}(\mathrm{D}\, V\, \mathrm{D}\, W).$$

Inserting everything into (A.1) and rearranging leads to the very compact and elegant representation of the shape Hessian for a volume integral:

$$\begin{aligned}\mathrm{d}^2\, J[V,W] = \int_\Omega & g \cdot (\operatorname{div}(V) \operatorname{div}(W) - \operatorname{tr} (\mathrm{D}\, V\, \mathrm{D}\, W)) \\ & + \mathrm{d}_{\mathrm{M}}\, g[V] \operatorname{div}(W) + \mathrm{d}_{\mathrm{M}}\, g[W] \operatorname{div}(V) + \mathrm{d}_{\mathrm{M}}^2\, g[V,W] \;\mathrm{d}\, x.\end{aligned} \tag{A.2}$$

The shape derivative procedure can now be applied to the material derivative of the Lagrangian in (5.3), which will then lead to the KKT system in variational form, ready to be implemented using finite elements.

$$\begin{aligned}&\mathrm{d}\, \mathscr{L}(u,\Omega,\lambda)[\mathrm{d}_{\mathrm{M}}\, u[V], V, \mathrm{d}_{\mathrm{M}}\, \lambda[V]] \\ &\quad = \int_\Omega \operatorname{div}(V) \left(\frac{1}{2}(u-z)^2 + \mu \langle \nabla u, \nabla \lambda \rangle - \lambda f \right) \\ &\qquad - (u-z) \mathrm{d}_{\mathrm{M}}\, z[V] - \lambda \mathrm{d}_{\mathrm{M}}\, f[V] \\ &\qquad - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T) \nabla \lambda \right\rangle + \langle \nabla u, \nabla \lambda \rangle \, \mathrm{d}_{\mathrm{M}}\, \mu[V] \\ &\qquad + (u-z) \mathrm{d}_{\mathrm{M}}\, u[V] + \langle \nabla \mathrm{d}_{\mathrm{M}}\, u[V], \nabla \lambda \rangle + \langle \nabla u, \nabla \mathrm{d}_{\mathrm{M}}\, \lambda[V] \rangle - f \mathrm{d}_{\mathrm{M}}\, \lambda[V] \;\mathrm{d}\, x \\ &\qquad + \alpha \int_{\partial \Omega_0} \operatorname{div}_\Gamma V \,\mathrm{d}\, s.\end{aligned} \tag{A.3}$$

The shape derivative for the perimeter regularization term is cited from [35]. Applying (A.2) to (A.3) will thus create an excessive amount of terms. To keep the derivation somewhat readable, we split (A.2) into separate terms,

$$\begin{aligned}
T_1[V,W] &:= g \cdot (\operatorname{div}(V)\operatorname{div}(W) - \operatorname{tr}(\mathrm{D}\,V\,\mathrm{D}\,W)),\\
T_2[V,W] &:= \mathrm{d_M}\, g[V]\operatorname{div}(W),\\
T_3[V,W] &:= \mathrm{d_M^2}\, g[V,W].
\end{aligned}$$

The KKT system can then be written as

$$\mathrm{d_M^2}\,\mathscr{L}[V,W] = \int_\Omega T_1[V,W] + T_2[V,W] + T_2[W,V] + T_3[V,W]\ \mathrm{d}\,x + \alpha \mathrm{d_M^2}\, R(\Omega)[V,W].$$

From the derivation of the gradient, we recall that in this special setting, we have

$$g = \frac{1}{2}(u-z)^2 + \mu\,\langle \nabla u, \nabla\lambda\rangle - \lambda f$$

and

$$\begin{aligned}
\mathrm{d_M}\, g[V] = &-(u-z)\mathrm{d_M}\, z[V] - \lambda \mathrm{d_M}\, f[V] - \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle\\
&+ (u-z)\mathrm{d_M}\, u[V] + \mu\,\langle \nabla \mathrm{d_M}\, u[V], \nabla\lambda\rangle + \mu\,\langle \nabla u, \nabla \mathrm{d_M}\,\lambda[V]\rangle\\
&- f\mathrm{d_M}\,\lambda[V].
\end{aligned}$$

We now have to formally derive $\mathrm{d_M^2}\, g[V,W] := \mathrm{d_M}\,(\mathrm{d_M}\, g[V])[W]$, which again generates an excessive amount of terms. To keep everything readable, we split the above second material derivative into even smaller parts, namely

$$\begin{aligned}
T_{3,2} &:= -(u-z)\mathrm{d_M}\, z[V] - \lambda \mathrm{d_M}\, f[V]\\
\Rightarrow \mathrm{d_M}\, T_{3,2}[W] &= -(\mathrm{d_M}\, u[W] - \mathrm{d_M}\, z[W])\mathrm{d_M}\, z[V] - (u-z)\mathrm{d_M^2}\, z[V,W]\\
&\quad - \mathrm{d_M}\,\lambda[W]\mathrm{d_M}\, f[V] - \lambda \mathrm{d_M^2}\, f[V,W]
\end{aligned}$$

and

$$T_{3,3} := \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle,$$

and the material derivative of this expression is computed as follows:

$$\begin{aligned}
\mathrm{d_M}\, T_{3,3}[W] &= -\mu\left\langle \mathrm{d_M}\,\nabla u[W], (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle - \mu\left\langle \nabla u, \mathrm{d_M}\left((\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right)[W]\right\rangle\\
&= -\mu\left\langle \nabla \mathrm{d_M}\, u[W] - \mathrm{D}\,W^T\nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle\\
&\quad - \mu\left\langle \nabla u, \mathrm{d_M}\left(\mathrm{D}\,V + \mathrm{D}\,V^T\right)[W]\nabla\lambda\right\rangle - \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\mathrm{d_M}\,(\nabla\lambda)[W]\right\rangle\\
&= -\mu\left\langle \nabla \mathrm{d_M}\, u[W], (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle + \mu\left\langle \mathrm{D}\,W^T\nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle\\
&\quad - \mu\left\langle \nabla u, (\mathrm{D}\,\mathrm{d_M}\,V[W] - \mathrm{D}\,V\,\mathrm{D}\,W + \mathrm{D}\,\mathrm{d_M}\,V[W]^T - (\mathrm{D}\,V\,\mathrm{D}\,W)^T)\nabla\lambda\right\rangle\\
&\quad - \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla \mathrm{d_M}\,\lambda[W]\right\rangle + \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\,\mathrm{D}\,W^T\nabla\lambda\right\rangle\\
&= -\mu\left\langle \nabla \mathrm{d_M}\, u[W], (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle + \mu\left\langle \mathrm{D}\,W^T\nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla\lambda\right\rangle\\
&\quad + \mu\left\langle \nabla u, (\mathrm{D}\,V\,\mathrm{D}\,W + (\mathrm{D}\,V\,\mathrm{D}\,W)^T)\nabla\lambda\right\rangle\\
&\quad - \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\nabla \mathrm{d_M}\,\lambda[W]\right\rangle + \mu\left\langle \nabla u, (\mathrm{D}\,V + \mathrm{D}\,V^T)\,\mathrm{D}\,W^T\nabla\lambda\right\rangle,
\end{aligned}$$

where we have again used $\mathrm{d_M}\, V[W] = 0$ in the last step. This object can be simplified further:

$$\begin{aligned}
\mathrm{d_M}\, T_{3,3}[W] =& -\mu \left\langle \nabla \mathrm{d_M}\, u[W], (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla\lambda \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla \mathrm{d_M}\, \lambda[W] \right\rangle \\
&+ \mu \left\langle \nabla u, \mathrm{D}\, W(\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla\lambda \right\rangle + \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T)\, \mathrm{D}\, W^T \nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, V\, \mathrm{D}\, W + (\mathrm{D}\, V\, \mathrm{D}\, W)^T)\nabla\lambda \right\rangle \\
=& -\mu \left\langle \nabla \mathrm{d_M}\, u[W], (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla\lambda \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla \mathrm{d_M}\, \lambda[W] \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V + \mathrm{D}\, W\, \mathrm{D}\, V^T)\nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, V\, \mathrm{D}\, W^T + (\mathrm{D}\, W\, \mathrm{D}\, V)^T)\nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, V\, \mathrm{D}\, W + (\mathrm{D}\, V\, \mathrm{D}\, W)^T)\nabla\lambda \right\rangle \\
=& -\mu \left\langle \nabla \mathrm{d_M}\, u[W], (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla\lambda \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T)\nabla \mathrm{d_M}\, \lambda[W] \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V + \mathrm{D}\, V\, \mathrm{D}\, W)\nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V^T + \mathrm{D}\, V\, \mathrm{D}\, W^T)\nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V)^T + (\mathrm{D}\, V\, \mathrm{D}\, W)^T)\nabla\lambda \right\rangle.
\end{aligned}$$

The material derivative of the next component is again straightforward:

$$\begin{aligned}
T_{3,4} &:= (u - z)\mathrm{d_M}\, u[V], \\
\mathrm{d_M}\, T_{3,4}[W] &= \mathrm{d_M}\, ((u - z)\mathrm{d_M}\, u[V])[W] \\
&= (\mathrm{d_M}\, u[W] - \mathrm{d_M}\, z[W])\mathrm{d_M}\, u[V] + (u - z)\mathrm{d_M}\, (\mathrm{d_M}\, u[V])[W].
\end{aligned}$$

The term $\mathrm{d_M}\, (\mathrm{d_M}\, u[V])[W]$ belongs to the purely nonsymmetric part of the second shape derivative and is left out in the linear second shape derivative. Thus, instead of $\mathrm{d_M}\, T_{3,4}[W]$, we use later on

$$\mathrm{d_M}\, T_{3,4}^{\mathrm{lin}}[W] = (\mathrm{d_M}\, u[W] - \mathrm{d_M}\, z[W])\mathrm{d_M}\, u[V].$$

The second to last expression is given by

$$T_{3,5} := \mu \left\langle \nabla \mathrm{d_M}\, u[V], \nabla\lambda \right\rangle + \mu \left\langle \nabla u, \nabla \mathrm{d_M}\, \lambda[V] \right\rangle$$

and the material derivative is computed as follows:

$$\begin{aligned}
\mathrm{d_M}\, T_{3,5}[W] =& \mu \left\langle \mathrm{d_M}\, (\nabla \mathrm{d_M}\, u[V])[W], \nabla\lambda \right\rangle + \mu \left\langle \nabla \mathrm{d_M}\, u[V], \mathrm{d_M}\, (\nabla\lambda)[W] \right\rangle \\
&+ \mu \left\langle \mathrm{d_M}\, (\nabla u)[W], \nabla \mathrm{d_M}\, \lambda[V] \right\rangle + \mu \left\langle \nabla u, \mathrm{d_M}\, (\nabla \mathrm{d_M}\, \lambda[V])[W] \right\rangle \\
=& \mu \left\langle \nabla \left(\mathrm{d_M}\, (\mathrm{d_M}\, u[V])[W]\right) - \mathrm{D}\, W^T \nabla \mathrm{d_M}\, u[V], \nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla \mathrm{d_M}\, u[V], \nabla \left(\mathrm{d_M}\, \lambda[W]\right) - \mathrm{D}\, W^T \nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla \left(\mathrm{d_M}\, u[W]\right) - \mathrm{D}\, W^T \nabla u, \nabla \mathrm{d_M}\, \lambda[V] \right\rangle \\
&+ \mu \left\langle \nabla u, \nabla \left(\mathrm{d_M}\, (\mathrm{d_M}\, \lambda[V])[W]\right) - \mathrm{D}\, W^T \nabla \mathrm{d_M}\, \lambda[V] \right\rangle.
\end{aligned}$$

Again, we ignore the nonsymmetric contributions $\mathrm{d_M}\, (\mathrm{d_M}\, u[V])[W]$ and $\mathrm{d_M}\, (\mathrm{d_M}\, \lambda[V])[W]$ in the changed term,

$$\begin{aligned}
\mathrm{d_M}\, T_{3,5}^{\mathrm{lin}}[W] =& -\mu \left\langle \mathrm{D}\, W^T \nabla \mathrm{d_M}\, u[V], \nabla\lambda \right\rangle + \mu \left\langle \nabla \mathrm{d_M}\, u[V], \nabla \left(\mathrm{d_M}\, \lambda[W]\right) - \mathrm{D}\, W^T \nabla\lambda \right\rangle \\
&+ \mu \left\langle \nabla \left(\mathrm{d_M}\, u[W]\right) - \mathrm{D}\, W^T \nabla u, \nabla \mathrm{d_M}\, \lambda[V] \right\rangle - \mu \left\langle \nabla u, \mathrm{D}\, W^T \nabla \mathrm{d_M}\, \lambda[V] \right\rangle.
\end{aligned}$$

This expression can also be simplified:

$$\begin{aligned}
\mathrm{d_M}\, T_{3,5}^{\mathrm{lin}}[W] =& -\mu \left\langle \nabla \mathrm{d_M}\, u[V], (\mathrm{D}\, W + \mathrm{D}\, W^T)\nabla\lambda \right\rangle + \mu \left\langle \nabla \mathrm{d_M}\, u[V], \nabla \mathrm{d_M}\, \lambda[W] \right\rangle \\
&+ \mu \left\langle \nabla \mathrm{d_M}\, u[W], \nabla \mathrm{d_M}\, \lambda[V] \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, W + \mathrm{D}\, W^T)\nabla \mathrm{d_M}\, \lambda[V] \right\rangle.
\end{aligned}$$

Last, but not least, we have

$$\begin{aligned} T_{3,6} &:= -f \mathrm{d}_{\mathrm{M}} \lambda[V], \\ \mathrm{d}_{\mathrm{M}} T_{3,6}[W] &= -\mathrm{d}_{\mathrm{M}} f[W] \mathrm{d}_{\mathrm{M}} \lambda[V] - f \mathrm{d}_{\mathrm{M}} (\mathrm{d}_{\mathrm{M}} \lambda[V])[W] \end{aligned}$$

and thus

$$\mathrm{d}_{\mathrm{M}} T_{3,6}^{\mathrm{lin}}[W] = -\mathrm{d}_{\mathrm{M}} f[W] \mathrm{d}_{\mathrm{M}} \lambda[V].$$

Taking everything together, we arrive at a variational expression for the shape-KKT system, ready to be implemented using mixed finite elements,

$$\begin{aligned} &\mathscr{L}''[\mathrm{d}_{\mathrm{M}} u[V], V, \mathrm{d}_{\mathrm{M}} \lambda[V]][\mathrm{d}_{\mathrm{M}} u[W], W, \mathrm{d}_{\mathrm{M}} \lambda[W]] \\ &= \int_\Omega \left(\frac{1}{2}(u-z)^2 + \mu \langle \nabla u, \nabla \lambda \rangle - \lambda f \right) (\operatorname{div}(V) \operatorname{div}(W) - \operatorname{tr}(\mathrm{D}\, V\, \mathrm{D}\, W)) \\ &\quad + \left(-(u-z)\mathrm{d}_{\mathrm{M}} z[V] - \lambda \mathrm{d}_{\mathrm{M}} f[V] - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T) \nabla \lambda \right\rangle \right) \operatorname{div} W \\ &\quad + \left(-(u-z)\mathrm{d}_{\mathrm{M}} z[W] - \lambda \mathrm{d}_{\mathrm{M}} f[W] - \mu \left\langle \nabla u, (\mathrm{D}\, W + \mathrm{D}\, W^T) \nabla \lambda \right\rangle \right) \operatorname{div} V \\ &\quad + ((u-z)\mathrm{d}_{\mathrm{M}} u[V] + \mu \langle \nabla \mathrm{d}_{\mathrm{M}} u[V], \nabla \lambda \rangle + \mu \langle \nabla u, \nabla \mathrm{d}_{\mathrm{M}} \lambda[V] \rangle - f \mathrm{d}_{\mathrm{M}} \lambda[V]) \operatorname{div} W \\ &\quad + ((u-z)\mathrm{d}_{\mathrm{M}} u[W] + \mu \langle \nabla \mathrm{d}_{\mathrm{M}} u[W], \nabla \lambda \rangle + \mu \langle \nabla u, \nabla \mathrm{d}_{\mathrm{M}} \lambda[W] \rangle - f \mathrm{d}_{\mathrm{M}} \lambda[W]) \operatorname{div} V \\ &\quad + \mathrm{d}_{\mathrm{M}} z[W] \mathrm{d}_{\mathrm{M}} z[V] - (u-z)\mathrm{d}_{\mathrm{M}}^2 z[V, W] \\ &\quad + \mathrm{d}_{\mathrm{M}} u[W] \mathrm{d}_{\mathrm{M}} u[V] - \mathrm{d}_{\mathrm{M}} u[V] \mathrm{d}_{\mathrm{M}} z[W] - \mathrm{d}_{\mathrm{M}} u[W] \mathrm{d}_{\mathrm{M}} z[V] \\ &\quad - \mathrm{d}_{\mathrm{M}} \lambda[W] \mathrm{d}_{\mathrm{M}} f[V] - \mathrm{d}_{\mathrm{M}} \lambda[V] \mathrm{d}_{\mathrm{M}} f[W] - \lambda \mathrm{d}_{\mathrm{M}}^2 f[V, W] \\ &\quad - \mu \left\langle \nabla \mathrm{d}_{\mathrm{M}} u[W], (\mathrm{D}\, V + \mathrm{D}\, V^T) \nabla \lambda \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, V + \mathrm{D}\, V^T) \nabla \mathrm{d}_{\mathrm{M}} \lambda[W] \right\rangle \\ &\quad - \mu \left\langle \nabla \mathrm{d}_{\mathrm{M}} u[V], (\mathrm{D}\, W + \mathrm{D}\, W^T) \nabla \lambda \right\rangle - \mu \left\langle \nabla u, (\mathrm{D}\, W + \mathrm{D}\, W^T) \nabla \mathrm{d}_{\mathrm{M}} \lambda[V] \right\rangle \\ &\quad + \mu \langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V + \mathrm{D}\, V\, \mathrm{D}\, W) \nabla \lambda \rangle \\ &\quad + \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V^T + \mathrm{D}\, V\, \mathrm{D}\, W^T) \nabla \lambda \right\rangle \\ &\quad + \mu \left\langle \nabla u, (\mathrm{D}\, W\, \mathrm{D}\, V)^T + (\mathrm{D}\, V\, \mathrm{D}\, W)^T) \nabla \lambda \right\rangle \\ &\quad + \mu \langle \nabla \mathrm{d}_{\mathrm{M}} u[W], \nabla \mathrm{d}_{\mathrm{M}} \lambda[V] \rangle + \mu \langle \nabla \mathrm{d}_{\mathrm{M}} u[V], \nabla \mathrm{d}_{\mathrm{M}} \lambda[W] \rangle \ \mathrm{d}\, x \\ &\quad + \alpha \int_{\partial \Omega_0} \operatorname{div}_\Gamma V \operatorname{div}_\Gamma W - \operatorname{tr}(\mathrm{D}_\Gamma V\, \mathrm{D}_\Gamma W) + \left\langle (\mathrm{D}_\Gamma V)^T n, (\mathrm{D}_\Gamma W)^T n \right\rangle \ \mathrm{d}\, s. \end{aligned}$$

The Hessian for the perimeter regularization term is cited from [35]. The term $\mathrm{d}_{\mathrm{M}}^2 z[V, W] = V^\top \mathrm{Hess}(z) W$ has to be assembled using finite elements. Often z itself is just a finite element approximation. If the order of this approximation is not higher than 1, then the assembled term is zero. Furthermore, this expression can also be a source of nonconvexity away from the optimal solution. Furthermore $\mathrm{d}_{\mathrm{M}}^2 f[V, W] = 0$ since the right-hand side f is assumed to deform with the mesh.

Acknowledgment. The authors are indebted to two anonymous referees, whose comments helped to improve the paper significantly.

REFERENCES

[1] L. Afraites, M. Dambrine, and D. Kateb, *On second order shape optimization methods for electrical impedance tomography*, SIAM J. Control Optim., 47 (2008), pp. 1556–1590.
[2] G. Allaire, *Conception optimale de structures*, Springer, New York, 2007.
[3] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, *The FEniCS project version* 1.5, Arch. Numer. Softw., 3 (2015), https://doi.org/10.11588/ans.2015.100.20553.

[4] S. Arguillère, E. Trélat, A. Trouvé, and L. Younes, *Shape deformation analysis from the optimal control viewpoint*, J. Math. Pures Appl., 104 (2015), pp. 139–178.
[5] E. Bängtsson, D. Noreland, and M. Berggren, *Shape optimization of an acoustic horn*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 1533–1571.
[6] 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, Cham, 2010, pp. 25–39,
[7] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, Philadelphia, 2012.
[8] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.
[9] M. Delfour and J.-P. Zolésio, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, Adv. Des. Control 22, 2nd ed., SIAM, Philadelphia, 2011.
[10] M. C. Delfour and J.-P. Zolésio, *Shapes and Geometries: Analysis, Differential Calculus, and Optimization,* Adv. Des. Control 4, SIAM, Philadelphia, 2001.
[11] P. Deuflhard, *Newton Methods for Nonlinear Problems, Affine Invariance and Adaptive Algorithms*, Springer Ser. Comput. Math. 35, Springer, Berlin, 2004.
[12] P. Deuflhard and G. Heindl, *Affine invariant convergence theorems for Newton's method and extensions to related methods*, SIAM J. Numer. Anal., 16 (1979), pp. 1–10.
[13] K. Eppler and H. Harbrecht, *A regularized Newton method in electrical impedance tomography using shape Hessian information*, Control. Cybernet., 34 (2005), pp. 203–225.
[14] K. Eppler and H. Harbrecht, *Shape optimization for free boundary problems – analysis and numerics*, in Constrained Optimization and Optimal Control for Partial Differential Equations, Internat. Ser. Numer. Math. 160, G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, and S. Ulbrich, eds., Birkhäuser, Basel, 2012, pp. 277–288.
[15] K. Eppler, H. Harbrecht, and R. Schneider, *On convergence in elliptic shape optimization*, SIAM J. Control. Optim., 46 (2007), pp. 61–83.
[16] K. Eppler, S. Schmidt, V. Schulz, and N. Gauger, *Preconditioning the pressure tracking in fluid dynamics by shape Hessian information*, J. Optim. Theory Appl., 141 (2009), pp. 513–531.
[17] I. Gherman, *Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems*, Ph.D. thesis, University of Trier, 2007.
[18] W. Gong and S. Zhu, *On discrete shape gradients of boundary type for PDE-constrained shape optimization*, SIAM J. Numer. Anal., 59 (2021), pp. 1510–1541.
[19] C. Groetsch, *Generalized Inverses of Linear Operators*, Marcel Dekker, New York, 1977.
[20] J. Haslinger and R. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation, and Computation*, Adv. Des. Control 7, SIAM, Philadelphia, 2003.
[21] K. Hayami, *Convergence of the Conjugate Gradient Method on Singular Systems,* NII-2018-001e, National Institute of Informatics, Tokyo, 2018.
[22] M. Hintermüller, *Fast level set based algorithms using shape and topological sensitivity information*, Control Cybernet., 34 (2005), pp. 305–324.
[23] R. Hiptmair, A. Paganini, and S. Sargheini, *Comparison of approximate shape gradients*, BIT, 55 (2015), pp. 459–485.
[24] A. Jameson, *Aerodynamic design via control theory*, J. Sci. Comput., 3 (1988), pp. 233–260.
[25] M. Laumen, *Newton's method for a class of optimal shape design problems*, SIAM J. Optim., 10 (1996), pp. 503–533.
[26] D. Luft and V. Schulz, *Pre-shape calculus: Foundations and application to mesh quality optimization*, Control Cybernet., 50 (2021), pp. 263–301.
[27] D. Luft and V. Schulz, *Simultaneous shape and mesh quality optimization using pre-shape calculus*, Control Cybernet., 50 (2021), pp. 473–520.
[28] A. Micheletti, *Metrica per famiglie di domini limitati e proprietá generiche degli autovalori*, Ann. Sc. Norm. Super. Pisa, 26 (1972), pp. 683–694.
[29] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*, Numer. Math. Sci. Comput., Clarendon Press, Oxford, 2001.
[30] F. Murat and J. Simon, *Etudes de problems d'optimal design*, in Optimization Techniques Modeling and Optimization in the Service of Man Part 2, Lecture Notes in Comput. Sci. 41, Springer, New York, 1976, pp. 54–62.
[31] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 2006.
[32] A. Novruzi and M. Pierre, *Structure of shape derivatives*, J. Evol. Equ., 2 (2002), pp. 365–382.
[33] S.-C. T. Choi, C. C. Paige, and M. A. Saunders, *MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems*, SIAM J. Sci. Comput., 33 (2011), pp. 1810–1836.

[34] S. Schmidt, M. Grässer, and H.-J. Schmid, *A Shape Newton Scheme for Deforming Shells with Application to Capillary Bridges*, https://arxiv.org/abs/2012.15016, 2020.
[35] S. Schmidt, M. Grässer, and H.-J. Schmid, *A shape Newton scheme for deforming shells with application to capillary bridges*, SIAM J. Sci. Comput., 44 (2022), pp. B1175–B1194, https://doi.org/10.1137/20M1389054.
[36] S. Schmidt, C. Ilic, V. Schulz, and N. Gauger, *Three-dimensional large-scale aerodynamic shape optimization based on shape calculus*, AIAA J., 51 (2013), pp. 2615–2627.
[37] V. Schulz, *A Riemannian view on shape optimization*, Found. Comput. Math., 14 (2014), pp. 483–501.
[38] V. Schulz and I. Gherman, *One-shot methods for aerodynamic shape optimization*, in MEGADESIGN and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design, Notes Numer. Fluid Mech. Multidiscip. Des. 107, N. Kroll, D. Schwamborn, K. Becker, H. Rieger, and F. Thiele, eds., Springer, New York, 2009, pp. 207–220.
[39] J. Sokolowski, *Displacement Derivatives in Shape Optimization of Thin Shells*, Research Report RR-2995, INRIA, 1996, https://hal.inria.fr/inria-00073702.
[40] J. Sokolowski and J. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer Ser. Comput. Math. 16, Springer, Berlin, 2012, https://books.google.de/books?id=bkruCAAAQBAJ.
[41] J. Sokolowski and J.-P. Zolésio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer, Berlin, 1992.
[42] K. Sturm, *Minimax Lagrangian approach to the differentiability of nonlinear PDE constrained shape functions without saddle point assumption*, SIAM J. Control. Optim., 53 (2015), pp. 2017–2039.
[43] K. Sturm, *Convergence of Newton's Method in Shape Optimisation via Approximate Normal Functions*, https://arxiv.org/abs/1608.02699, 2018.
[44] K. Welker, *Suitable spaces for shape optimization*, Appl. Math. Optim., 84 (2021), pp. S869–S902, https://doi.org/10.1007/s00245-021-09788-2.
[45] L. Younes, *Shapes and Diffeomorphisms*, Appl. Math. Sci. 171, 2nd ed., Springer, Berlin, 2019.
[46] S. Zhu, *Effective shape optimization of laplace eigenvalue problems using domain expressions of eulerian derivatives*, J. Optim. Theory Appl., 176 (2018), pp. 17–34.