THE STRIP METHOD FOR SHAPE DERIVATIVES

SEAN HARDESTY, HARBIR ANTIL, DREW P. KOURI, AND DENIS RIDZAL

Abstract. A major challenge in shape optimization is the coupling of finite element method (FEM) codes in a way that facilitates efficient computation of shape derivatives. This is particularly difficult with multiphysics problems involving legacy codes, where the costs of implementing and maintaining shape derivative capabilities are prohibitive. The volume and boundary methods are two approaches to computing shape derivatives. Each has a major drawback: the boundary method is less accurate, while the volume method is more invasive to the FEM code. We introduce the *strip method*, which computes shape derivatives on a strip adjacent to the boundary. The strip method makes code coupling simple. Like the boundary method, it queries the state and adjoint solutions at quadrature nodes, but requires no knowledge of the FEM code implementations. At the same time, it exhibits the higher accuracy of the volume method. As an added benefit, its computational complexity is comparable to that of the boundary method, i.e., it is faster than the volume method. We illustrate the benefits of the strip method with numerical examples.

1. Introduction

In shape optimization, the question of whether to discretize-then-optimize or to optimize-then-discretize is closely related to the choice of method used for evaluating shape gradients. There are two equivalent mathematical formulations of the shape gradient: the volume method (which requires the evaluation of a volume integral) and the boundary method (which requires the evaluation of a surface integral). A relationship between these two approaches was established in [6], where it was shown that the volume method, when combined with a finite element method (FEM) discretization, amounts to shape differentiation of FEM matrix entries. Consquently, the volume method is extremely difficult to implement and maintain in legacy FEM codes. More recently, the authors of [11, 12, 16] proved that the volume method achieves a higher order of accuracy than the boundary method in most cases. In

Key words and phrases. Shape Optimization, Strip Method, Volume Method, Boundary Method, Finite Elements .

SH was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories. HA was partially supported by the U.S. Air Force Office of Scientific Research, Optimization Program under Award NO: FA9550-19-1-0036, the Department of Navy, under Award NO: N00244-20-1-0005, and NSF grants DMS-1818772, DMS-1913004. DPK and DR were supported by the U.S. Air Force Office of Scientific Research, Optimization Program under Award NO: F4FGA09135G001.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

this paper, we propose a method that recovers the accuracy of the volume method and that can be added to legacy FEM codes in a nonintrusive manner.

As demonstrated by the Hadamard Structure Theorem [8, Ch. 9, Th. 3.6], shape gradients are fully supported on the boundary of the domain. Upon discretizing the volume method shape gradient using FEM, one would expect the values at interior mesh nodes to decrease with the mesh size. This observation was alluded to in [19, §5], where the authors state that "nonzero integral contributions [in the interior of the domain] are caused by discretization noise". As a result, the method described in [19] restricts the computations to the boundary in order to maintain stability of Hessian approximations. Using this as motivation, we propose the *strip method*. Roughly speaking, the strip method maintains a separate FEM discretization on a subset of the computational domain that contains the boundary (the "strip"). This strip is then used to compute shape derivatives via the volume method.

There are numerous computational benefits to the strip method. Most importantly, the strip method is not invasive to the existing FEM code: it only requires the state and adjoint solutions evaluated at quadrature points. So long as the strip method uses a mesh that conforms to the boundary of the existing FEM mesh, it need not use the same mesh in the strip, nor even the same element topology. The strip mesh can be refined independently of the existing FEM mesh (see Section 5.2), making it possible to implement the strip method with lower-order elements while retaining the accuracy of the volume method in a high-order FEM code. The strip method could even be implemented with automatic differentiation (AD). In AD, code tools are used to differentiate the discretization rather than taking the approach of the volume method, which is to discretize the shape gradient. AD is a promising method for shape differentiation [15], but without the strip method, it suffers from the same invasiveness as the volume method.

Aside from the computational benefits of the strip method, a critical question that must be addressed in shape optimization is how does one update the underlying mesh as the shape is modified. Common mesh updating schemes from the literature include:

- Remesh at every step [21] – a straightforward choice, although it has been suggested that this approach risks numerical noise [4, §4.3];
- Conformal mapping [13] – no mesh update is needed, but it only works in two dimensions;
- Interface FEM [9] – implicit local modification of the FEM mesh in the vicinity of interfaces, without actually moving mesh nodes;
- Shell Methods [3, 2, 10] – there is no interior to update because shell equations are used rather than elasticity over a volume;
- Harmonic Mapping [14, 18] – mapping of the boundary deformation to the interior via solution of an elliptic boundary value problem (sometimes with additional ad hoc smoothing used in the interior);
- Do Nothing [5] – exemplary of many papers where the question of changes to the interior mesh simply is not addressed, and where it is not clear that such changes are even necessary given that changes to the surface shape are sufficiently small.

None of these approaches advocate the use of the volume method gradient to update the interior mesh vertices. The fact that the volume method gradient is a poor means of updating

the interior mesh is confirmed in our numerical experiments, in Section 5. As a consequence, the strip method does not discard any useful information.

The remainder of the paper is organized as follows. In Section 2, we introduce our model shape optimization problem and expressions for the shape gradients. In Section 3, we describe the strip method. This is followed by Section 4, where we discuss three approaches to approximate the shape gradients on the boundary: the boundary method, the volume method and the strip method, demonstrating that FEM error estimates for the volume method can be extended to the case of the strip method. We conclude with several illustrative numerical examples in Section 5. Implementation details are provided in B.

2. Problem Formulation and Background

We consider the classical PDE-constrained shape optimization problem [8, 17, 20]. Let $\Omega \subset \mathbb{R}^N$, with $1 \leq N \leq 3$, be a bounded open set with a sufficiently smooth boundary $\partial\Omega$. We wish to solve the problem

$$\min_{\Omega} \mathcal{J}(\Omega) := \int_{\Omega} j(u)\, dx, \tag{2.1a}$$

where $\mathcal{J}$ is the shape functional, the integrand $j : \mathbb{R} \to \mathbb{R}$ has a locally Lipschitz continuous derivative j_u, and u in (2.1a) solves the PDE

$$\begin{cases} -\Delta u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \partial\Omega. \end{cases} \tag{2.1b}$$

We assume that $f \in L^2(\mathbb{R}^N)$ and we identify f with its restriction on Ω. Throughout, we refer to (2.1b) as the state equation. Moreover, we denote the topological dual space associated with a Banach space X by X' and we denote the associated duality pairing by $\langle \cdot, \cdot \rangle_{X',X}$.

2.1. Shape Gradients and the Structure Theorem.

In order to solve (2.1) using derivative-based optimization, we must establish the concept of a shape gradient. We model perturbations of Ω using the map

$$y_{\boldsymbol{V}} := \boldsymbol{I} + \boldsymbol{V}, \tag{2.2}$$

where $\boldsymbol{I}$ denotes the identity map and $\boldsymbol{V}$ is a vector field in $\mathcal{D}^1 := \mathcal{D}^1(\mathbb{R}^N; \mathbb{R}^N)$. The space $\mathcal{D}^1$ consists of functions that are continuously differentiable in $\mathbb{R}^N$ with compact support. Moreover, the map in (2.2) is a diffeomorphism provided $\|\boldsymbol{V}\|_{\mathcal{D}^1} < 1$. See, for instance, [1, Lemma 6.13] from which we also notice that compactly supported perturbation vector fields in $W^{1,\infty}(\mathbb{R}^N, \mathbb{R}^N)$ are sufficient for this mapping to be a diffeomorphism. However, since our exposition relies on some results from [12], we work with $\boldsymbol{V} \in \mathcal{D}^1$. We define the shape functional $\mathcal{J} : \Omega_{\text{ad}} \to \mathbb{R}$ on an admissible set of domains

$$\Omega_{\text{ad}} := \{ y_{\boldsymbol{V}}(\Omega) \ : \ \boldsymbol{V} \in \mathcal{D}^1, \ \|\boldsymbol{V}\|_{\mathcal{D}^1} < 1 \}.$$

We say that the shape functional $\mathcal{J}$ is Gâteaux differentiable at Ω if the limit

$$d\mathcal{J}(\Omega; \boldsymbol{V}) := \lim_{t \downarrow 0} \frac{\mathcal{J}(y_{t\boldsymbol{V}}(\Omega)) - \mathcal{J}(\Omega)}{t}$$

exists for all $\boldsymbol{V} \in \mathcal{D}^1$ and the map $d\mathcal{J}(\Omega;\cdot) : \mathcal{D}^1 \to \mathbb{R}$ is linear and bounded. The above map defines a vector distribution $\boldsymbol{G}_\Omega \in (\mathcal{D}^1)'$. Following the terminology from [8, Ch. 9, Def. 3.4(ii)], we call this distribution the *volume* shape gradient. Notice that in terms of $\boldsymbol{G}_\Omega$, we have

$$\langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = d\mathcal{J}(\Omega;\boldsymbol{V}) \quad \forall\, \boldsymbol{V} \in \mathcal{D}^1. \tag{2.3}$$

Let $\gamma_{\partial\Omega}(\cdot) : \mathcal{D}^1 \to C^1(\partial\Omega)$ be the trace operator. Then for C^2-domains Ω, we have that

$$\gamma_{\partial\Omega}(\boldsymbol{V}) \cdot \boldsymbol{\nu} \in C^1(\partial\Omega),$$

with $\boldsymbol{\nu}$ being the outward unit normal. The Hadamard Structure Theorem [8, Ch. 9, Th. 3.6] gives a representation of $\boldsymbol{G}_\Omega$ in (2.3) in terms of the *boundary* shape gradient $\boldsymbol{G}_{\partial\Omega} := G_{\partial\Omega}\boldsymbol{\nu} \in C^1(\partial\Omega)'$ as follows:

$$\langle \boldsymbol{G}_{\partial\Omega}, \gamma_{\partial\Omega}(\boldsymbol{V}) \rangle_{C^1(\partial\Omega)',C^1(\partial\Omega)} = \langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1}, \quad \text{for all } \boldsymbol{V} \in \mathcal{D}^1. \tag{2.4}$$

Namely, (2.4) establishes a direct relationship between the volume shape gradient $\boldsymbol{G}_\Omega$ and its representative $\boldsymbol{G}_{\partial\Omega}$ on the boundary $\partial\Omega$. Indeed, from (2.4), we have that

$$\boldsymbol{G}_\Omega = \gamma^*_{\partial\Omega}(\boldsymbol{G}_{\partial\Omega}), \tag{2.5}$$

where the adjoint of the trace operator $\gamma^*_{\partial\Omega}$ is well-defined due to the linearity and boundedness of $\gamma_{\partial\Omega}$.

2.2. **Shape Gradients for** (2.1)**.** The volume shape gradient of $\mathcal{J}$ for the PDE-constrained shape optimization problem (2.1) has the following form. Given a perturbation $\boldsymbol{V} \in \mathcal{D}^1$, the volume shape gradient applied to $\boldsymbol{V}$ is given by

$$\begin{aligned} \langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Omega \Big(&\nabla u \cdot \left(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top\right) \nabla p - f \boldsymbol{V} \cdot \nabla p \\ &+ \operatorname{div} \boldsymbol{V} \left(j(u) - \nabla u \cdot \nabla p\right) \Big)\, dx \end{aligned} \tag{2.6}$$

where p solves the adjoint equation

$$\begin{cases} -\Delta\, p &= j_u(u) \quad \text{in} \quad \Omega \\ \quad p &= 0 \qquad \text{on} \quad \partial\Omega. \end{cases} \tag{2.7}$$

We refer to [12, § 2] for a derivation of (2.6). See A for intuition regarding terms in (2.6).

Remark 2.1. *If $f \in H^1(\Omega)$, we can write the corresponding term in* (2.6) *as*

$$-\int_\Omega f \boldsymbol{V} \cdot \nabla p \, dx = \int_\Omega div \boldsymbol{V} p f \, dx + \int_\Omega p(\boldsymbol{V} \cdot \nabla f) \, dx,$$

where we have used that $p|_{\partial\Omega} = 0$. As a result, the shape derivative in (2.6)*, based on the volume integral, becomes*

$$\begin{aligned} \langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Omega \Big(&\nabla u \cdot \left(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top\right) \nabla p + p(\boldsymbol{V} \cdot \nabla f) \\ &+ div\, \boldsymbol{V} \left(j(u) - \nabla u \cdot \nabla p + pf\right) \Big)\, dx. \end{aligned} \tag{2.8}$$

Compared to (2.6)*, the expression* (2.8) *is more readily related to shape derivatives of discretizations of f. See the discussion following* (B.7) *in B.*

Next, we recall the expression for the boundary shape gradient in the direction $\boldsymbol{V}$,

$$\langle \boldsymbol{G}_{\partial\Omega}, \gamma_{\partial\Omega}(\boldsymbol{V})\rangle_{C^1(\partial\Omega)',C^1(\partial\Omega)} = \int_{\partial\Omega} (\boldsymbol{V}\cdot\boldsymbol{\nu})\,(j(u) + \partial_{\boldsymbol{\nu}} p \partial_{\boldsymbol{\nu}} u)\ d\sigma. \tag{2.9}$$

Here, $\boldsymbol{G}_{\partial\Omega} = G_{\partial\Omega}\boldsymbol{\nu} = (j(u) + \partial_{\boldsymbol{\nu}} p \partial_{\boldsymbol{\nu}} u)\,\boldsymbol{\nu}$. Using integration by parts, one can show that the volume directional derivative (2.6) and boundary directional derivative (2.9) are the same (see the Appendix of [12]), i.e., the Hadamard Structure Theorem (2.4) holds.

Both the volume and boundary methods require the solution of the state equation (2.1b) and the adjoint equation (2.7), after which either the volume expression (2.6) or the boundary expression (2.9) can be evaluated to compute the shape gradient. Ideally, one would implement discretizations of (2.6) and (2.9) in the original FEM code. However, this is often not possible when dealing with complicated discretizations and legacy codes. The boundary method is a natural option for these situations. In particular, the expression (2.9) depends on boundary data available from the FEM code and on geometric terms including the surface normal and mean surface curvature. A set of routines to handle these geometric terms would easily be extensible to a wide variety of problems; see [18, 20]. On the other hand, the main impediment to the volume method is that the integral (2.6) must be evaluated over the entire volume. Consequently, it is often impractical to add the required shape differentiation capabilities. In the next section, we introduce the strip method as a remedy for this complication. Similar to the boundary method, the strip method is minimally invasive to the FEM code. At the same time, it retains the higher accuracy of the volume method.

3. The Strip Method

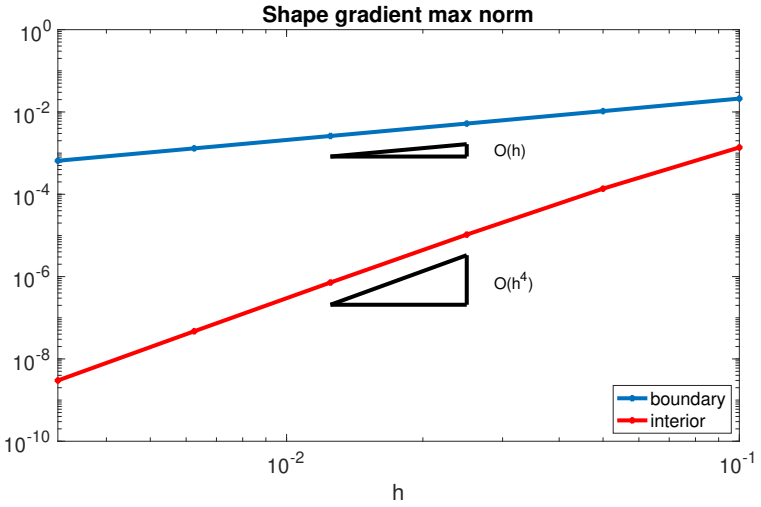

FIGURE 1. L^∞-norm of the discrete volume method gradient restricted to the boundary versus restricted to the interior as the mesh is refined for the example described in Section 5. Because the values on the interior are so small, updating the interior mesh using the volume method is not a useful strategy; see the results of this approach in Figure 8.

When discretizing (2.6) using FEM, the values of the discrete shape gradient on the boundary only receive contributions from elements touching the boundary. Moreover, as demonstrated in Figure 1, the values of the discrete shape gradient computed on the interior of the mesh tend to zero as the mesh is refined. This suggests that the important contributions to (2.6) only occur in the vicinity of the boundary. The Hadamard Structure Theorem provides theoretical justification for this observation as it states that the shape gradient has support contained in $\partial\Omega$. These observations motivate the strip method.

Let $\Sigma \subseteq \Omega$ be a fixed open set satisfying $\partial\Omega \subsetneq \partial\Sigma$. We refer to Σ as the strip and define the smooth cut-off function $\psi : \mathbb{R}^N \to [0,1]$ to be infinitely differentiable and satisfy

$$\psi(x) \in \begin{cases} \{1\} & \text{if } x \in \mathbb{R}^N \setminus \Omega \\ (0,1) & \text{if } x \in \Sigma \\ \{0\} & \text{if } x \in \Omega \setminus \Sigma \end{cases}.$$

Note that since ψ is infinitely differentiable, it satisfies $\nabla\psi(x) = 0$ for all $x \in \partial\Sigma$. The smooth cut-off functions ψ and $1-\psi$ define a smooth partition of unity that we use to decompose any $\boldsymbol{V} \in \mathcal{D}^1$ as

$$\boldsymbol{V} = \psi\boldsymbol{V} + (1-\psi)\boldsymbol{V}.$$

In the subsequent discussion, we will use Ψ to denote the continuous linear operator defined by $\boldsymbol{V} \mapsto \psi\boldsymbol{V} : \mathcal{D}^1 \to \mathcal{D}^1$. By the Hadamard Structure Theorem [8, Ch. 9, Th. 3.6], the shape gradient $\boldsymbol{G}_\Omega$ has support completely contained in $\partial\Omega$ and hence

$$\begin{aligned} \langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} &= \langle \boldsymbol{G}_\Omega, \psi\boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} + \langle \boldsymbol{G}_\Omega, (1-\psi)\boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} \\ &= \langle \boldsymbol{G}_\Omega, \Psi\boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1}. \end{aligned}$$

As a result, we define the strip-based shape derivative as

$$\begin{aligned} \langle \boldsymbol{G}_\Sigma, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} &= \langle \Psi^* \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} \\ &= \langle \boldsymbol{G}_\Omega, \Psi\boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} \quad \forall \boldsymbol{V} \in \mathcal{D}^1. \end{aligned} \tag{3.1}$$

For our problem, the strip shape gradient $\boldsymbol{G}_\Sigma$ in the direction $\boldsymbol{V}$ is

$$\begin{aligned} \langle \boldsymbol{G}_\Sigma, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Sigma \Big(&\nabla u \cdot \left(\nabla(\Psi\boldsymbol{V}) + \nabla(\Psi\boldsymbol{V})^\top \right) \nabla p - f(\Psi\boldsymbol{V}) \cdot \nabla p \\ &+ \operatorname{div}(\Psi\boldsymbol{V}) \left(j(u) - \nabla u \cdot \nabla p \right) \Big) \, dx, \end{aligned} \tag{3.2}$$

where (u,p) solve the state and adjoint equations, respectively.

4. Approximation of the Strip Shape Gradient

In [12], the authors derive error estimates for the volume and boundary shape gradients and their respective FEM approximations. In this section, we extend these error estimates to the strip method. To do this, we first introduce our FEM discretization for the state and adjoint equations. Let $\mathcal{T}_h = \{K\}$ be a conforming, shape-regular, and quasi-uniform triangulation of Ω, where $K \subset \mathbb{R}^N$ is an element that is isoparametrically equivalent to either the unit cube or the unit simplex in $\mathbb{R}^N$ and h denotes the element size. We denote

the collection of all such meshes by $\mathbb{T}$, i.e., $\mathbb{T} = \{\mathcal{T}_h\}$. For every $\mathcal{T}_h \in \mathbb{T}$, we define the finite-element space as

$$\mathbb{V}_h := \mathbb{V}(\mathcal{T}_h) = \left\{u_h \in C^0(\overline{\Omega}) \ : \ u_h|_K \in \mathcal{P}_1(K),\ u_h|_{\partial\Omega} = 0\right\}.$$

When K in the above definition is a simplex, then $\mathcal{P}_1 = \mathbb{P}_1(K)$, the set of polynomials of degree at most 1, while if K is a cube, then $\mathcal{P}_1 = \mathbb{Q}_1(K)$, the set of polynomials of partial degree at most 1.

To discretize both the volume and boundary methods, we substitute the approximations of the state and adjoint variables $(u, p) \in H_0^1(\Omega) \times H_0^1(\Omega)$ that solve (2.1b) and (2.7) into (2.6) and (2.9), respectively. We denote these approximations by $u_h \in \mathbb{V}_h$ and $p_h \in \mathbb{V}_h$, where u_h and p_h solve

$$\int_\Omega \nabla u_h \cdot \nabla v_h \, dx = \int_\Omega f v_h \, dx \quad \forall v_h \in \mathbb{V}_h, \tag{4.1}$$

$$\int_\Omega \nabla v_h \cdot \nabla p_h \, dx = \int_\Omega j_u(u_h) v_h \, dx \quad \forall v_h \in \mathbb{V}_h, \tag{4.2}$$

respectively. The (semi-)discretized representations of $\boldsymbol{G}_\Omega$ and $\boldsymbol{G}_{\partial\Omega}$ are then given by

$$\begin{aligned}\langle \boldsymbol{G}_\Omega^h, \boldsymbol{V}\rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Omega \Big(&\nabla u_h \cdot \left(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top\right) \nabla p_h - f \boldsymbol{V} \cdot \nabla p_h \\ &+ \operatorname{div} \boldsymbol{V} \left(j(u_h) - \nabla u_h \cdot \nabla p_h\right) \Big) \, dx\end{aligned} \tag{4.3}$$

and

$$\langle \boldsymbol{G}_{\partial\Omega}^h, \gamma_{\partial\Omega}(\boldsymbol{V})\rangle_{C^1(\partial\Omega)',C^(\partial\Omega)} = \int_{\partial\Omega} (\boldsymbol{V} \cdot \boldsymbol{\nu}) \left(j(u_h) + \partial_{\boldsymbol{\nu}} p_h \partial_{\boldsymbol{\nu}} u_h\right) \, d\sigma \, . \tag{4.4}$$

Due to the approximations (u_h, p_h), the Hadamard Structure Theorem need not apply, i.e., in general

$$\boldsymbol{G}_\Omega^h \neq \gamma_{\partial\Omega}^*(\boldsymbol{G}_{\partial\Omega}^h).$$

The following error estimates for the volume and boundary shape gradients with simplicial triangulation were derived in [12] and demonstrate that $\boldsymbol{G}_\Omega^h$ and $\boldsymbol{G}_{\partial\Omega}^h$ do not in general exhibit the same rate of convergence with mesh refinement.

Theorem 4.1. *Let Ω be convex or $C^{1,1}$ and f be the restriction of an $H^1(\mathbb{R}^N)$ function onto Ω. If $(u, p) \in H_0^1(\Omega) \times H_0^1(\Omega)$ solves* (2.1b) *and* (2.7)*, and $(u_h, p_h) \in \mathbb{V}_h \times \mathbb{V}_h$ solves* (4.1) *and* (4.2)*, then the following holds:*

$$\left|\langle \boldsymbol{G}_\Omega - \boldsymbol{G}_\Omega^h, \boldsymbol{V}\rangle_{(\mathcal{D}^1)',\mathcal{D}^1}\right| \leq C_1(\Omega, u, p, f) h^2 \|\boldsymbol{V}\|_{W^{2,4}(\Omega)}. \tag{4.5}$$

For an explicit dependence of C_1 on the underlying variables we refer to [12, Th. 3.1]*. In addition, if*

$$\|u\|_{W^{2,p}(\Omega)} \leq C \|f\|_{L^p(\Omega)}$$

for some $p > N$ and a positive constant C, then

$$\left|\langle \boldsymbol{G}_{\partial\Omega} - \boldsymbol{G}_{\partial\Omega}^h, \gamma_{\partial\Omega}(\boldsymbol{V})\rangle_{C^1(\partial\Omega)',C^1(\partial\Omega)}\right| \leq C_2 h \|\boldsymbol{V} \cdot \boldsymbol{\nu}\|_{L^\infty(\partial\Omega)} \tag{4.6}$$

where the constant $C_2 > 0$ is independent of h.

Similar to the approximation of the the volume shape gradient, we can approximate the strip shape gradient $\boldsymbol{G}_\Sigma$ in (3.2) by substituting (u, p) with (u_h, p_h). The (semi-)discretized strip shape gradient is

$$\begin{aligned}\langle \boldsymbol{G}_\Sigma^h, \boldsymbol{V}\rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Sigma \Big(&\nabla u_h \cdot \left(\nabla(\Psi\boldsymbol{V}) + \nabla(\Psi\boldsymbol{V})^\top\right)\nabla p_h - f(\Psi\boldsymbol{V})\cdot\nabla p_h \\ &+ \operatorname{div}\,(\Psi\boldsymbol{V})\left(j(u_h) - \nabla u_h\cdot\nabla p_h\right)\Big)\,dx.\end{aligned} \tag{4.7}$$

Note that $\boldsymbol{G}_\Sigma^h = \Psi^*\boldsymbol{G}_\Omega^h$, which leads to the following error estimate.

Corollary 4.2. *Let the assumptions of Theorem 4.1 hold, and let the smooth cut-off function Ψ define a continuous linear operator from $W^{2,4}(\Omega)$ into $W^{2,4}(\Omega)$. Then, we have*

$$\left|\langle \boldsymbol{G}_\Sigma - \boldsymbol{G}_\Sigma^h, \boldsymbol{V}\rangle_{(\mathcal{D}^1)',\mathcal{D}^1}\right| \le C_1(\Omega, u, p, f)h^2\|\Psi\|\|\boldsymbol{V}\|_{W^{2,4}(\Omega)}, \tag{4.8}$$

where C_1 is the constant in Theorem 4.1 and $\|\Psi\|$ denotes the operator norm of Ψ in $W^{2,4}(\Omega)$.

Proof. Since $\boldsymbol{G}_\Sigma = \Psi^*\boldsymbol{G}_\Omega$ and $\boldsymbol{G}_\Sigma^h = \Psi^*\boldsymbol{G}_\Omega^h$, Theorem 4.1 ensures that

$$\left|\langle \boldsymbol{G}_\Sigma - \boldsymbol{G}_\Sigma^h, \boldsymbol{V}\rangle_{(\mathcal{D}^1)',\mathcal{D}^1}\right| \le C_1(\Omega, u, p, f)h^2\|\Psi\boldsymbol{V}\|_{W^{2,4}(\Omega)}.$$

The result then follows from the assumed continuity of the smooth cut-off operator Ψ. □

The strip method is ideal for use with complicated discretizations and legacy codes. In particular, suppose u_h and p_h were computed using an arbitrary FEM discretization. To compute the strip shape gradient, one can maintain finite elements on the strip Σ that are potentially different from the finite elements for Ω (i.e., the mesh cells could have different topologies and the finite element basis functions could be of different orders). One can then evaluate the integral in (4.7) by querying the FEM interpolants u_h and p_h at quadrature points associated with the discretization on Σ. Additionally, all differentiation with respect to nodal coordinates of FEM quantities such as element Jacobians and shape functions only needs to be performed for the strip discretization. This allows the strip discretization to be refined to match the accuracy of the Ω discretization. To use the strip shape gradient for shape optimization, one possible strategy to update the Ω mesh is as follows.

(i) Define

$$\widetilde{\boldsymbol{G}}_{\partial\Omega}^h = \gamma_{\partial\Omega}(\boldsymbol{G}_\Sigma^h). \tag{4.9}$$

We emphasize that in (4.9), the trace of $\boldsymbol{G}_\Sigma^h$ is not the only possible choice. Instead, for instance, one could use the L^2-projection of $\boldsymbol{G}_\Sigma^h$ onto the $\partial\Omega$.

(ii) Provide the discrete gradient $\widetilde{\boldsymbol{G}}_{\partial\Omega}^h$ to the optimizer.

(iii) Update the interior mesh via harmonic extension of the boundary node perturbations produced by the optimizer.

In Section 5, we present a numerical example that contrasts this method of mesh updating with the approach of using the volume gradient $\boldsymbol{G}_\Omega^h$ to update the interior mesh.

5. Numerical Examples

We have discretized the optimization problem (2.1a)-(2.1b), choosing the domain shape, objective function, and boundary conditions in such a way as to make the problem challenging for the Boundary Method. Note that our model problem (2.1b) has Dirichlet boundary conditions, which is known from [12, § 4] to be the more challenging case.

We have chosen the initial domain to be $\Omega_0 = (0,1)^2$ and aim to recover the circle Ω_* that circumscribes Ω_0:

$$\Omega_* = \left\{ x \in \mathbb{R}^2 \ : \ \left| x - \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}^\top \right|_2 < \frac{\sqrt{2}}{2} \right\}.$$

Turning the square into a circle via optimization requires smoothing out its corners, which poses challenges for mesh quality.

The integrand j in the objective function (2.1a) is chosen to be

$$j(u) = \frac{1}{2}(u - u_*)^2, \tag{5.1}$$

where

$$u_*(x) = J_0\left(\lambda \left| x - \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}^\top \right|_2\right)$$

solves (2.1b) on Ω_* with forcing function

$$f(x) = \lambda^2 u_*(x).$$

In the above, J_0 is the order-zero Bessel function of the first kind, and λ is chosen so that the first zero of J_0 occurs at the boundary of the circle, i.e., the smallest positive real number such that

$$J_0\left(\lambda \frac{\sqrt{2}}{2}\right) = 0.$$

Remark 5.1. *Note that the form of* (5.1) *differs from that of* (2.1a) *originally assumed in our theoretical development: the addition of u_* introduces an explicit dependence of j on x, i.e., $j(u(x), x)$ instead of $j(u(x))$. All that is needed to handle this in the discretization is the additional term $\int_\Omega \nabla_x j(u(x), x) \cdot \boldsymbol{V}$ in the gradient expressions* (2.6) *and* (2.8)*; see* [20, § 2.16].

First, we consider the convergence of the gradient as the mesh is refined, both in approximate dual norms (Section 5.1), and in the case of this particular example where it makes sense, in $L^2(\partial\Omega)$ (Section 5.2). Additionally, we consider two approaches to solving the discretized optimization problem. The first (Section 5.3) uses a single code, whereas the second (Section 5.4) uses two decoupled codes, as outlined in Section 3.

5.1. **Strip Refinement in Approximate Dual Norms.** In this section, we apply the technique proposed by Hiptmair et al. [12, § 4] for investigating convergence of discrete shape gradients. To summarize, we compute the operator norm over $\boldsymbol{V} \in \mathcal{P}_{3,3}(\mathbb{R}^2)$, and replace the $C^1(\mathbb{R}^N; \mathbb{R}^N)$-norm with the $H^1(\Omega)$-norm. This simplifies the computations, while still making it possible to demonstrate $O(h^2)$ convergence. Since the exact solution is unknown, we also compute the solution on a further-refined mesh and compute errors relative to it.

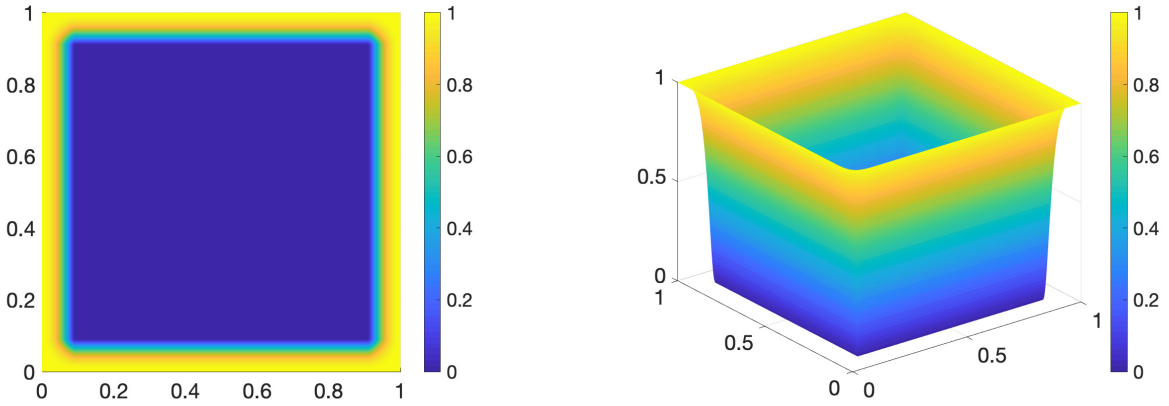

FIGURE 2. The smooth cutoff function for the strip, $\psi(x; 0.1)$; see (5.2)

Aside from using the example problem described in Section 5, the main difference between the strip method and the approach taken in Hiptmair et al. is that we pre-multiply the functions $\boldsymbol{V} \in \mathcal{P}_{3,3}(\mathbb{R}^2)$ by a smooth cutoff function. Let

$$\phi(\zeta) = 1 - \exp\left(1 - \frac{1}{1-\zeta^2}\right),$$

and for $0 < a < 1$,

$$c(\zeta; a) = \begin{cases} \phi(\zeta/a) & \zeta < a \\ \phi((1-\zeta)/a) & \zeta > 1 - a \\ 1 & \text{otherwise.} \end{cases}$$

Then, our cutoff function is defined via

$$\psi(x; a) = 1 - c(x_1; a)c(x_2; a). \tag{5.2}$$

Its support is contained in $\Sigma \subset \Omega$, where Ω is the unit square, and

$$\Sigma = \{x \in \Omega : x_1 < a \text{ or } x_1 > 1 - a \text{ or } x_2 < a \text{ or } x_2 > 1 - a\};$$

see Figure 2. As in Section 3, the construction of ψ allows us to compute $\langle \boldsymbol{G}_\Sigma, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1}$ as $\langle \boldsymbol{G}_\Omega, \Psi \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1}$; see (3.1). In Figure 3, the volume and strip shape gradients are shown to converge at the same rate.

This approach demonstrates that we achieve the theoretical rate of convergence from Theorem 4.1 and Corollary 4.2. However, in order to solve the optimization problem using gradient-based methods, we need to evaluate the shape gradient of the objective function, and not just the directional derivatives. Furthermore, we must consider the relation between discrete gradients computed on different meshes in order to implement the strip method with the volume mesh and the strip mesh being independently refined. We address these issues in the following subsection.

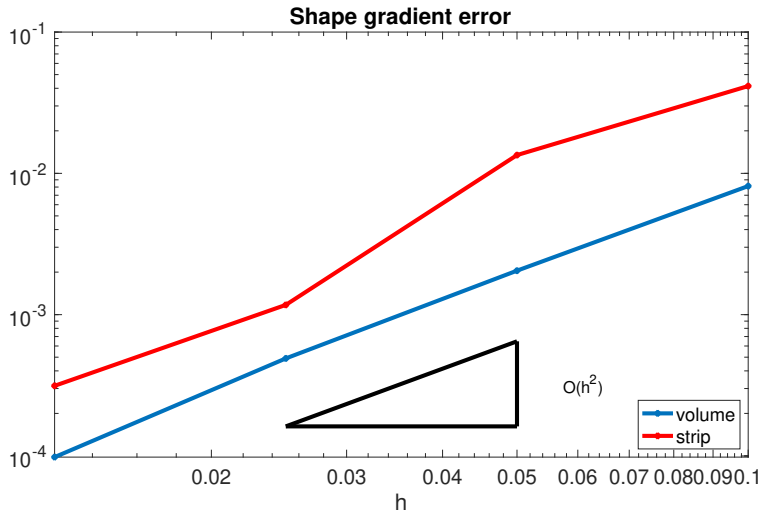

FIGURE 3. Convergence in the approximate dual norms for the volume and strip methods. Both achieve the expected $O(h^2)$ rate. The strip method exhibits somewhat higher error until the mesh is refined sufficiently for the element quadrature rules to integrate the rapidly-changing ψ accurately.

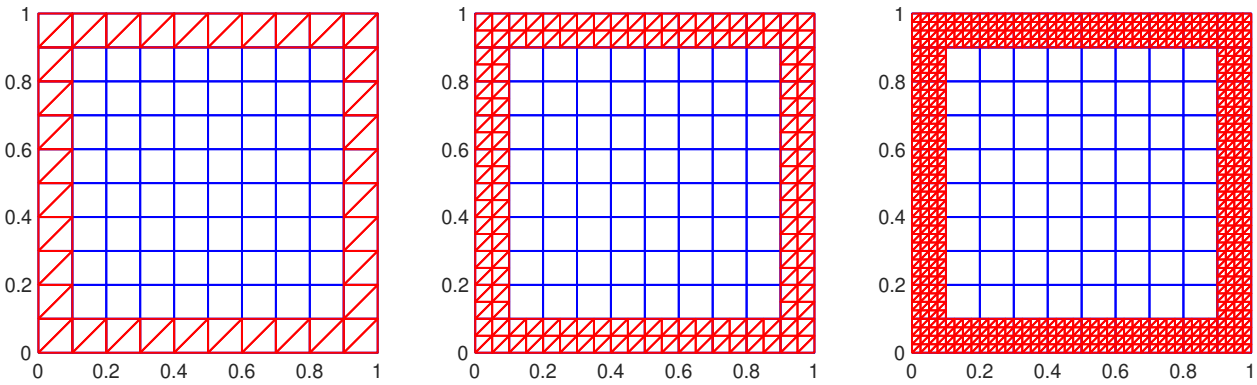

FIGURE 4. From left to right: successive levels of refinement of the strip mesh.

5.2. **Strip Refinement in $L^2(\partial\Omega)$.** In this subsection, we consider independent refinement of the volume mesh, used for computation of the state and adjoint equations, and of the strip mesh, used for the shape gradient computation. The strip method is implemented with a quadrilateral mesh for the state and adjoint, and a triangular mesh of the strip, as shown in Figure 4. The volume method is implemented using the same triangular mesh for both calculations.

Lacking a closed-form expression for the gradient, we compute the discrete gradient on a much finer mesh and coarsen it back to the original mesh. The final comparison between the "exact" solution and the solution given by different levels of refinement in the strip is done in the L^2-norm; see Figure 5. For our example problem, the strip method exhibits $O(h^2)$ convergence, and has lower errors than the volume method. Up to a point, independent refinement of the strip mesh can reduce the error because the refined triangular mesh is better

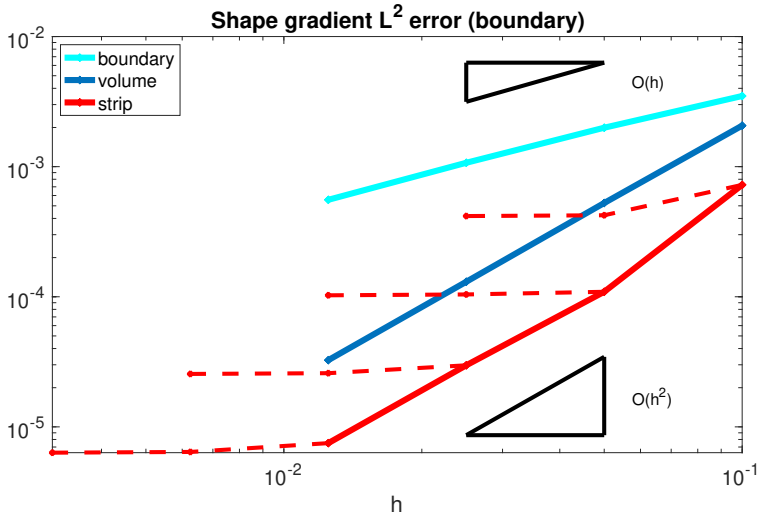

FIGURE 5. L^2 error in the shape gradient with mesh refinement. Since we are evaluating the improvements that can be made relative to a given discretization, all refined solutions are projected back to the coarsest mesh prior to error computation. The "exact" solution is computed on a much finer mesh. **In blue:** the volume method as its single mesh is uniformly refined. **In red:** the strip method, where each dashed curve indicates the sequence of mesh refinements shown in Figure 4. **In cyan:** the boundary method.

able to approximate the the quadrilateral mesh basis functions. We anticipate a greater benefit for independent refinement in the case where the underlying FEM discretization is of higher order.

Next, we outline the method by which the discrete gradient is coarsened. Consider a coarse mesh $\mathcal{M}_c$ with Lagrange basis functions $\{\varphi_i^c : i = 1, \dots, N_c\}$, and a finer mesh $\mathcal{M}_f$ with Lagrange basis functions $\{\psi_i^f : i = 1, \dots, N_f\}$. Consider, e.g., the volume representation (2.8):

$$\langle \boldsymbol{G}_\Omega, \boldsymbol{V} \rangle_{(\mathcal{D}^1)',\mathcal{D}^1} = \int_\Omega \Big(\nabla u \cdot \left(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top \right) \nabla p + p(\boldsymbol{V} \cdot \nabla f) + \operatorname{div} \boldsymbol{V} \left(j(u) - \nabla u \cdot \nabla p + pf \right) \Big) \, dx.$$

If

$$\boldsymbol{V}(x) = \begin{pmatrix} \sum_{i=1}^{N_c} \varphi_i^c(x) v_{1,i}^c \\ \sum_{i=1}^{N_c} \varphi_i^c(x) v_{2,i}^c \end{pmatrix},$$

then the discretization of (2.8) would have components

$$g^c_{1,i} = \int_\Omega \left(\nabla u \cdot \begin{pmatrix} 2\partial_1\varphi^c_i(x) & \partial_2\varphi^c_i(x) \\ \partial_2\varphi^c_i(x) & 0 \end{pmatrix} \nabla p + p \left(\begin{pmatrix} \varphi^c_i(x) \\ 0 \end{pmatrix} \cdot \nabla f \right) \right.$$
$$\left. + \partial_1\varphi^c_i(x) \left(j(u) - \nabla u \cdot \nabla p + pf \right) \right) dx,$$
$$g^c_{2,i} = \int_\Omega \left(\nabla u \cdot \begin{pmatrix} 0 & \partial_1\varphi^c_i(x) \\ \partial_1\varphi^c_i(x) & 2\partial_2\varphi^c_i(x) \end{pmatrix} \nabla p + p \left(\begin{pmatrix} 0 \\ \varphi^c_i(x) \end{pmatrix} \cdot \nabla f \right) \right.$$
$$\left. + \partial_2\varphi^c_i(x) \left(j(u) - \nabla u \cdot \nabla p + pf \right) \right) dx.$$

Naturally, as the mesh is refined, the entry $g^c_{\alpha,i}$ ($\alpha = 1$ or 2) corresponding to a particular node becomes smaller because the support of φ^c_i shrinks. Therefore, simple interpolation of the gradient vectors is not correct. To translate to a coarser mesh, we require that if a displacement field $v^c_\alpha = \begin{pmatrix} v^c_{\alpha,1} & v^c_{\alpha,2} & \cdots & v^c_{\alpha,N_c} \end{pmatrix}^\top$ of the coarse mesh is applied to the fine mesh, then the translation of the fine gradient should compute the same value:

$$\left(v^c_\alpha\right)^\top \hat{g}^c_\alpha = \left(v^f_\alpha\right)^\top g^f_\alpha. \tag{5.3}$$

Here, g^f_α is known, and we are trying to compute an "equivalent" $\hat{g}^c_\alpha$.

We match the variations v^c_α and v^f_α via L^2-projection, i.e., by minimizing

$$\int_\Sigma \left(\sum_{i=1}^{N_c} \varphi^c_i(x) v^c_{\alpha,i} - \sum_{i=1}^{N_c} \psi^f_j(x) v^f_{\alpha,j} \right)^2 dx.$$

Given the mass matrices $M_{\mathrm{ff}} \in \mathbb{R}^{N_f \times N_f}$ and $M_{\mathrm{fc}} \in \mathbb{R}^{N_f \times N_c}$, with entries

$$\left(M_{\mathrm{ff}}\right)_{ij} = \int_\Sigma \psi^f_i(x) \psi^f_j(x)\, dx$$
$$\left(M_{\mathrm{fc}}\right)_{ij} = \int_\Sigma \psi^f_i(x) \varphi^c_j(x)\, dx,$$

we obtain the condition

$$M_{\mathrm{ff}} v^f_\alpha = M_{\mathrm{fc}} v^c_\alpha. \tag{5.4}$$

Combining (5.4) and (5.3),

$$\left(v^c_\alpha\right)^\top \hat{g}^c_\alpha = \left(v^c_\alpha\right)^\top M_{\mathrm{fc}}^\top M_{\mathrm{ff}}^{-1} g^f_\alpha,\ \forall v^c_\alpha,$$

so we obtain the simple coarsening formula

$$\hat{g}^c_\alpha = M_{\mathrm{fc}}^\top M_{\mathrm{ff}}^{-1} g^f_\alpha.$$

5.3. **Single Code.** In the single code approach to the discretized optimization problem, the objective function (2.1a), the state equation (2.1b), the adjoint equation (2.7), and the shape gradient expressions (2.8) and (2.9) are all discretized using the same triangular mesh with piecewise-linear basis functions. In this case, the volume method computes the objective function gradient to machine precision (see Figure 6), but must be implemented within the underlying FEM code. This section compares the behavior of the volume method and the boundary method in the case of a single code.

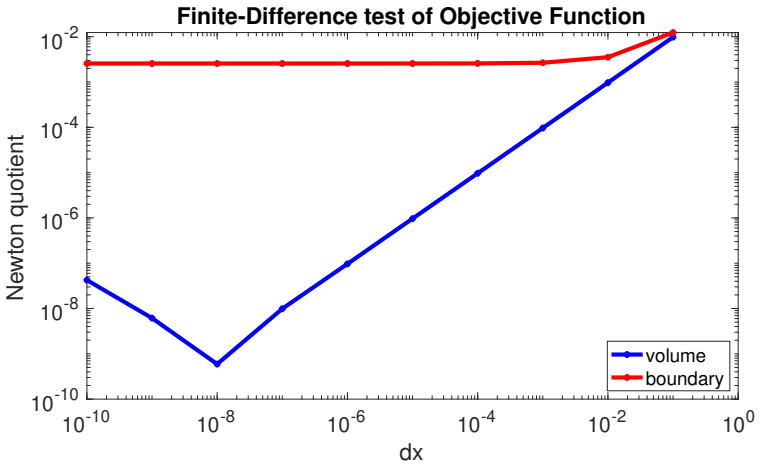

FIGURE 6. Numerical verification of the correctness of the implementation. Here are plotted Newton quotients of the form $(f(x+\delta)-f(x)-\nabla f(x)\delta)/|\delta|_{\ell^2}$ versus a step size $|\delta|_{\ell^2}$. We obtain good agreement down to about $\sqrt{\epsilon_{\text{mach}}}$ for the volume method because, as shown in B, discretization of the volume method is equivalent to differentiation of the finite element matrix entries. The Newton quotients flat-line for the boundary method due to the non-equivalence of $\boldsymbol{G}_{\Omega}^{h}$ and $\boldsymbol{G}_{\partial\Omega}^{h}$.

Recall that at the end of Section 4, we outlined a strategy based on harmonic extension for updating the interior mesh as the boundary mesh moves during optimization. The results of this approach are shown in Figure 7. In the remainder of this section, we perform optimization without this harmonic update of the interior mesh. The optimizer can achieve substantially greater reduction of the objective function and gradient values without the harmonic mesh update. This occurs because the gradient we compute does not incorporate the effects of the changing discretization implied by the mesh motion. Naturally, this additional term could be included in the gradient, but we wish to understand the differences in the behavior of optimization algorithms using the volume and boundary methods without this added complication.

In Figure 8, we show optimization results without the harmonic mesh update (cf. Figure 7). At first, the gradient is dominated by the shape changes of the boundary (Figure 8**(a)**). Once the boundary shape is resolved this approach changes the discretization error by moving the interior nodes in a way that further reduces the objective function, at the expense of mesh quality (Figure 8**(b)**). If the interior mesh nodes are not moved at all, then the optimizer can be run safely until convergence (Figure 8**(c)**).

As a result of these observations, the comparison with the boundary method presented in Figure 9 uses the volume method gradient to perturb the boundary vertices only. After about 10 iterations, the volume method reaches an objective function value comparable to the final objective value attained by the boundary method after 30 iterations. Moreover, the volume method continues to reduce the objective function until it is approximately three times smaller. Since the boundary method gradient is inconsistent with the discretization, we choose to stop at stagnation of the method rather than using a gradient tolerance.

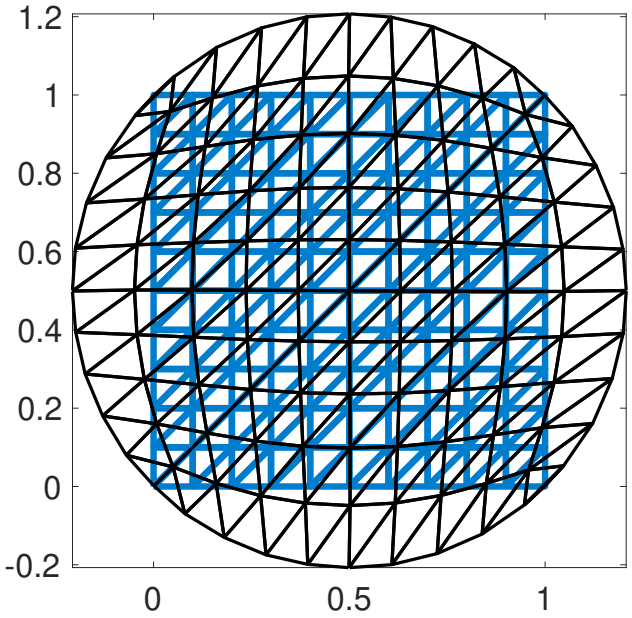

FIGURE 7. Optimized mesh using the volume gradient to update boundary mesh nodes and harmonic extension to update interior mesh nodes (black) overlaid on the original mesh (blue). Objective function and gradient values are 1.1720e-06, 5.6347e-06. **N.B.** The elements beginning at the corners $x = (0,1)^\top$ and $x = (1,0)^\top$, are deformed nearly out of existence. Although this problem could be mitigated with an irregular mesh, it speaks to the limitations of harmonic mapping as a strategy for the maintenance of mesh quality.

It is also interesting to make some observations about the behavior of the volume method gradient on the interior of the domain. These are shown in Figure 1. The $O(h)$ behavior on the boundary is expected for piecewise linear elements: the scale of entries of the discrete gradient vector must decrease in proportion to the size of the elements as the mesh is refined. Interestingly, the entries on the interior scale approximately as $O(h^4)$. We don't have theory to explain this rate, and cannot speak to its generality having run it only on the example described in this section. The general behavior is consistent with the observations in [19, §5]: when a small perturbation is made to an interior node, it effectively changes the discretization error, so the volume method in this case is differentiating the discretization error.

5.4. **Decoupled Codes.** In this subsection, two separate FEM codes are used together on the discretized optimization problem. The first, Intrelab [7], plays the role of the legacy code, and is used to discretize the objective function (2.1a), the state equation (2.1b), and the adjoint equation (2.7) using bilinear quadrilateral elements. The second code discretizes the shape gradient expressions (3.2) and (2.9) using piecewise linear triangular elements. This section compares the behavior of the strip method and the boundary method implemented in separate codes.

The strip method retains the accuracy of the volume method, but this is only apparent in a finite-difference test when both Intrelab and the shape gradient code use the same mesh; see Figure 10. The accuracy of the strip method can be increased via refinement of the

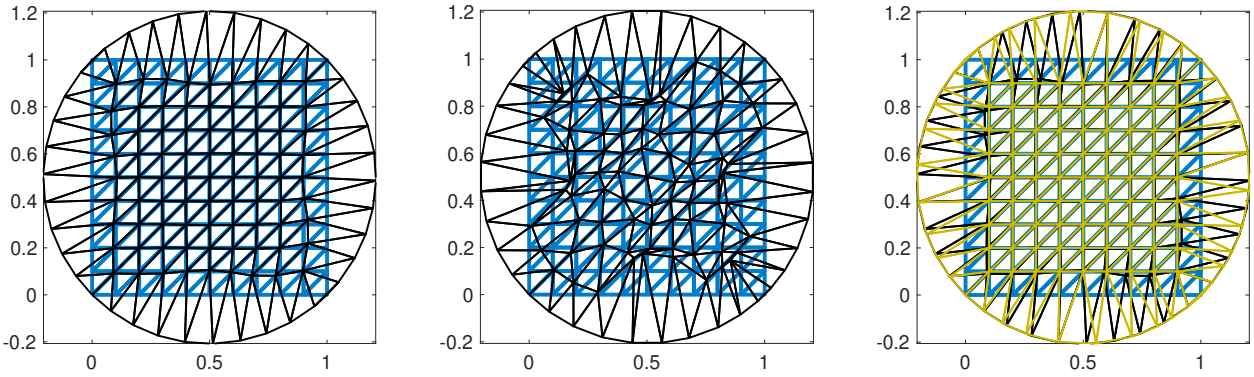

FIGURE 8. From left to right: **(a)** Optimized mesh using the volume gradient to update all mesh nodes (black) overlaid on the original mesh (blue). This example is run until the infinity-norm of the gradient reaches 5.6347e-06 (the level achieved in Figure 7), at which point the step size tolerance has not yet been reached, and significant further decrease in the objective function is still possible. Objective function and gradient values are 1.9673e-06, 3.9946e-06. The interior nodes move very little, consistent with the observations in Figure 1. **(b)** As in (a), but run to a step size of 10^{-10}. The optimizer takes advantage of the consistency of the gradient discretization (see B) to reduce the objective function by an additional factor of 10^4: objective function and gradient values are 2.4905e-10, 1.2782e-08. The distortion of interior elements begins after the boundary shape is already well-resolved. **(c)** Same as (a), but including (yellow) the results of running the volume method to a step size of 10^{-10} and allowing the optimizer to perturb only the boundary vertices. Objective function and gradient values are 1.1031e-06, 9.7883e-07. It is interesting to note that these two approaches result in quite different distributions of nodes on the target circle.

triangular mesh on the strip; see Section 5.2. After about 10 iterations, the volume method reaches an objective function value comparable to the final objective value attained by the boundary method after 23 iterations; see Figure 11. We note that the boundary method appears to perform better than in Figure 9. This may be due to a more accurate evaluation of the Neumann data in (2.9), using bilinear quadrilateral elements.

6. Conclusion

As observed in [19], discretizations of the volume method produce negligibly small values on interior nodes. Consequently, these values are not useful in updating the mesh as the domain boundary is moved. In addition, the volume method is often difficult to add to legacy FEM codes, while the boundary method results in lower-accuracy gradient computations. To address these observations, we have introduced the strip method. The strip method computes a volume shape gradient on a strip near the boundary of the domain and can be discretized using a mesh and elements that are different from those used by the underlying FEM code to compute the state and adjoint. As a result, the computational complexity

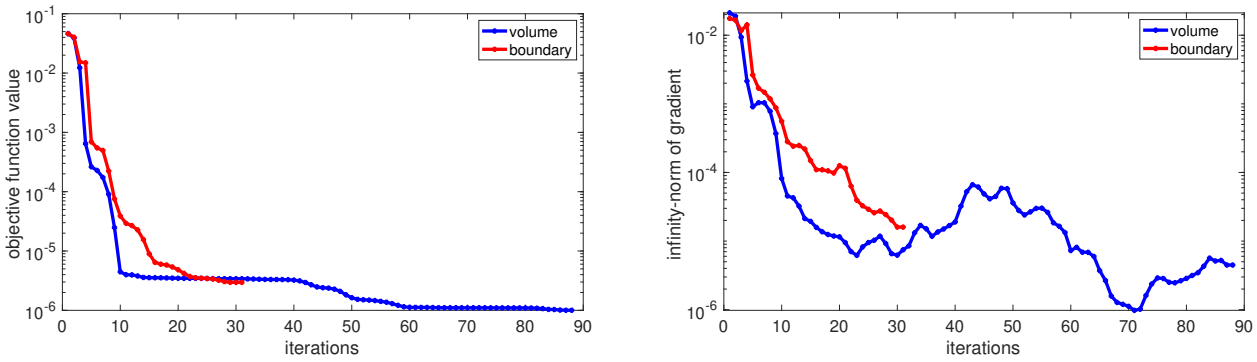

FIGURE 9. Convergence of the objective function and gradient values (∞-norm) for both the volume and boundary methods. The stopping criterion is a minimum step size of 10^{-10}. The volume method reaches the objective function value achieved by the boundary method much faster, and subsequently continues to reduce it, by a factor of three.

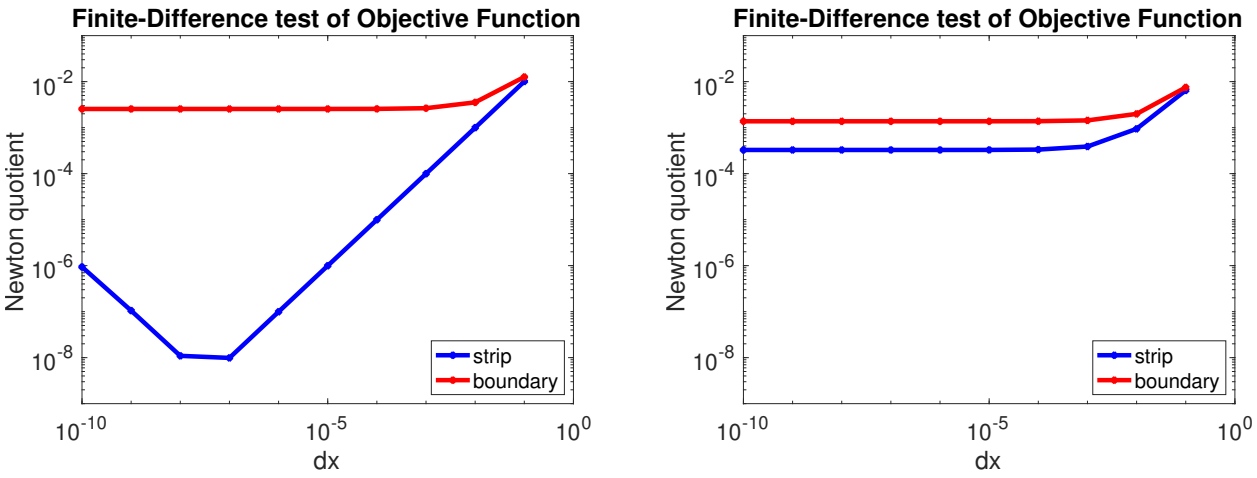

FIGURE 10. Numerical verification of the correctness of the implementation. Here are plotted Newton quotients of the form $(f(x+\delta)-f(x)-\nabla f(x)\delta)/|\delta|_{\ell^2}$ versus a step size $|\delta|_{\ell^2}$; compare with Figure 6. From left to right: **(a)** Intrelab using triangular elements, strip using triangular elements. This demonstrates that the discretizations are consistent. **(b)** Intrelab using quadrilateral elements, strip using triangular elements. In this case, the error with the strip method flat-lines because the objective function value and its gradient are computed using inconsistent discretizations. This does not imply that the gradient computed using quadrilateral elements for the state and adjoint is less accurate; see Section 5.2. Note also that the boundary method errors are reduced from those in (a) because the boundary data is computed with bilinear quadrilaterals instead of linear triangles.

of the strip method can be made comparable to that of the boundary method and one can recover the accuracy of the volume method by refining the strip mesh. We have demonstrated

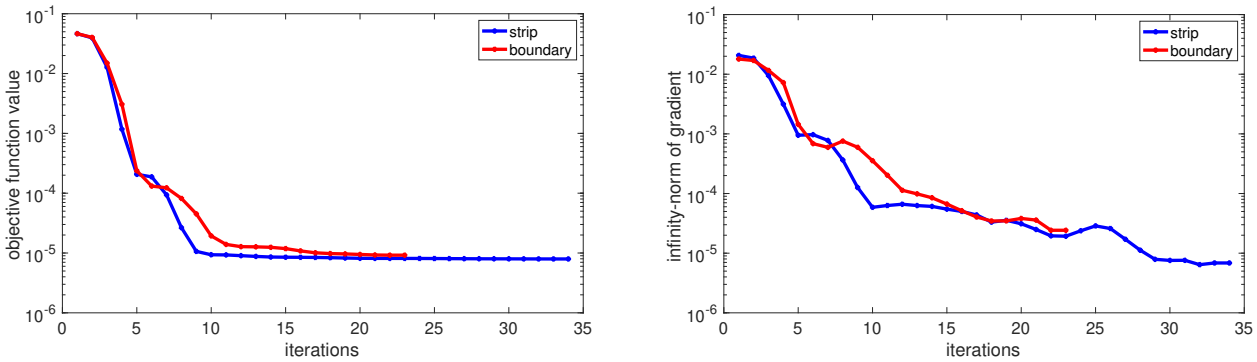

FIGURE 11. Convergence of the objective function and gradient values (∞-norm) for both the strip and boundary methods. The stopping criterion is a minimum step size of 10^{-10}. The strip method reaches the objective function value achieved by the boundary method much faster. The performance of the boundary method appears better than in Figure 9. This may be due to the fact that the Neumann data required in the boundary method expression (2.9) is computed more accurately by the bilinear quadrilateral elements.

this approach in Section 5, where we showed that convergence for our model optimization problem occurs in about the same number of iterations as when implemented using the usual volume shape gradient. Finally, the strip method makes it possible to use AD tools without the challenges of integrating them into legacy codes. Only mass matrices are required for independent mesh refinement, computed as described in Section 5.2. Altogether, the strip method combines the best in speed, accuracy, and implementation cost from among the existing methods of shape gradient computation, while simplifying code coupling.

APPENDIX A. VOLUME SHAPE GRADIENT INTUITION

In this appendix, we provide some intuition regarding the terms $\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top$ and div $\boldsymbol{V}$ in the volume shape gradient (2.6). Recall the mapping $\Omega \ni x \to y(x) := x + \boldsymbol{V}(x)$ from (2.2), where for simplicity we have dropped the subscript $\boldsymbol{V}$ from y. The bilinear form associated with the Laplacian in (2.1b) over the perturbed domain $y(\Omega)$ is given by

$$\int_{y(\Omega)} \nabla u \cdot \nabla v \, dy = \int_{\Omega} \nabla u(y(x))^\top \nabla y(x) \nabla y(x)^\top \nabla v(y(x)) \det(\nabla y(x)) \, dx.$$

Since $\nabla y(x) = I + \nabla \boldsymbol{V}$, we have that

$$\nabla y \nabla y^\top = I + \nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top + \nabla \boldsymbol{V} \nabla \boldsymbol{V}^\top$$

from which we obtain the term corresponding to $\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^\top$ in equation (2.6). Likewise, the change of volume term $\det(\nabla y(x))$ evaluates to

$$1 + \text{div } \boldsymbol{V} + O(|\nabla \boldsymbol{V}|^2),$$

hence the origin of the div $\boldsymbol{V}$ term in (2.6).

Appendix B. Implementation Details

Discretization of the boundary shape gradient $\boldsymbol{G}_{\partial\Omega}$ in (2.9) to obtain $\boldsymbol{G}^h_{\partial\Omega}$ in (4.4) is quite straightforward with piecewise-linear finite elements. The complexity of the boundary method increases with more complex PDE models or with more challenging boundary conditions [20].

In this section, we focus on the discretization of the volume shape gradient $\boldsymbol{G}_\Omega$ (or $\boldsymbol{G}_\Sigma$). We shall consider two approaches: (i) discretize-then-optimize, where we first discretize the optimization problem and then evaluate the discrete volume shape gradient; (ii) optimize-then-discretize, where we discretize $\boldsymbol{G}_\Omega$ in (2.8) (or $\boldsymbol{G}_\Sigma$). We will show that, for this example, the two approaches are equivalent. In doing so, we aim to repeat the analysis of [6] in the standard language of FEM matrix assembly. We emphasize that the boundary representative of the discrete volume shape gradient $\boldsymbol{G}^h_\Omega$, denoted by $\widetilde{\boldsymbol{G}}^h_{\partial\Omega}$ is computed by simply taking the trace of $\boldsymbol{G}^h_\Omega$ (or $\boldsymbol{G}^h_\Sigma$).

For simplicity of notation, we restrict the discussion below to 2-D. However, the same ideas can be applied in higher dimensions. Let $\{\varphi_i : i = 1, \dots, N_h\}$ be a nodal Lagrange basis for $\mathbb{V}_h$. Then with

$$u_h(x) = \sum_{i=1}^{N_h} U_i \varphi_i(x), \quad p_h(x) = \sum_{i=1}^{N_h} P_i \varphi_i(x),$$

the discrete state and adjoint equations (4.1) and (4.2) can be written

$$KU = F \tag{B.1}$$

$$K^\top P = G, \tag{B.2}$$

where

$$\begin{aligned} K_{ij} &= \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, dx, \\ F_i &= \int_\Omega f\varphi_i \, dx, \\ G_i &= \int_\Omega j_{u_h}(u_h)\varphi_i \, dx. \end{aligned} \tag{B.3}$$

Assume that there are S_h nodal coordinates, which in two dimensions is $S_h = 2N_h$, corresponding to shape variables s_k, $k \in \{1, \dots, S_h\}$. We will denote, e.g., the discrete shape derivative of K for $k \in \{1, \dots, S_h\}$ by

$$(K')_{ijk} = \frac{\partial K_{ij}}{\partial s_k}.$$

Using this notation, the discrete shape derivative of (B.1) is

$$K'U + KU' = F',$$

which can be rearranged into the discrete shape sensitivity equation

$$U' = K^{-1}\left(F' - K'U\right),$$

The discrete volume shape gradient is then

$$G^\top U' = \underbrace{G^\top K^{-1}}_{=P^\top}(F' - K'U)\,, \tag{B.4}$$

where we recall from (B.2) that $P^\top = G^\top K^{-1}$. To complete the description, we need to work out formulas for the discrete operator derivatives K' and F' in (B.4). It is sufficient to consider their local assembly, i.e., on a single triangle. Recall that the local contributions to K and F are given in (B.3).

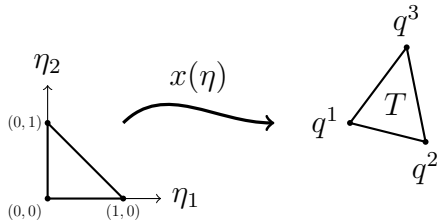

FIGURE 12. Mapping from unit triangle reference element with coordinates $\eta \in \mathbb{R}^2$ to a triangle T with vertices $q^1, q^2, q^3 \in \mathbb{R}^2$.

The mapping of the reference element to a mesh triangle with vertices $q^1, q^2, q^3 \in \mathbb{R}^2$ (see Figure 12) is given by

$$x(\eta) = q^1 + \underbrace{\begin{pmatrix} q^2 - q^1 & q^3 - q^1 \end{pmatrix}}_{\equiv J} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}. \tag{B.5}$$

We define the standard piecewise-linear basis functions via the mapping $\ell : \mathbb{R}^2 \to \mathbb{R}^3$ via

$$\ell(\eta) = \begin{pmatrix} 1 - \eta_1 - \eta_2 \\ \eta_1 \\ \eta_2 \end{pmatrix},$$

and express them in mesh coordinates via

$$\hat{\ell}(x(\eta)) = \ell(\eta).$$

We thus obtain the local quantities

$$\mathbb{R}^{3\times 3} \ni K_{\text{loc}} = \int_T \nabla\hat{\ell}\nabla\hat{\ell}^\top\, dx = \int_0^1 \int_0^{1-\eta_1} \nabla\ell J^{-1} J^{-\top} \nabla\ell^\top \det J\, d\eta_2 d\eta_1, \tag{B.6}$$

$$\mathbb{R}^{3\times 1} \ni F_{\text{loc}} = \int_T \hat{\ell} f(x)\, dx = \int_0^1 \int_0^{1-\eta_1} \ell f(x(\eta)) \det J\, d\eta_2 d\eta_1, \tag{B.7}$$

recalling the definition of the Jacobian matrix J in (B.5). Differentiation of K_{loc} with respect to the six local degrees of freedom (two per vertex) on the triangle T is thus a matter of differentiating the terms J^{-1} and $\det J$. Differentiation of F_{loc} requires differentiation of the the argument $x(\eta)$ in $f(x(\eta))$, and differentiation of $\det J$. These are respectively analogous to the terms $p(\boldsymbol{V}\cdot\nabla f)$ and $(\text{div}\,\boldsymbol{V})pf$ in (2.8).

Recall than J can be expressed (c.f. (B.5); the subscript denotes the first and second components)

$$J = \begin{pmatrix} q_1^2 - q_1^1 & q_1^3 - q_1^1 \\ q_2^2 - q_2^1 & q_2^3 - q_2^1 \end{pmatrix}, \tag{B.8}$$

so that

$$\det J = (q_1^2 - q_1^1)(q_2^3 - q_2^1) - (q_2^2 - q_2^1)(q_1^3 - q_1^1). \tag{B.9}$$

Next, we shall compute the shape variations of the $\det J$ and J^{-1}. Consider a perturbation $(\delta q^1, \delta q^2, \delta q^3)$ of the vertices (q^1, q^2, q^3) of an element T. The change in $\det J$ in (B.9) is given by

$$\begin{aligned}(\det J)'\delta q =&(\delta q_1^2 - \delta q_1^1)(q_2^3 - q_2^1) - (\delta q_2^2 - \delta q_2^1)(q_1^3 - q_1^1) + \\ &(q_1^2 - q_1^1)(\delta q_2^3 - \delta q_2^1) - (q_2^2 - q_2^1)(\delta q_1^3 - \delta q_1^1).\end{aligned}$$

Thus, if we arrange the local shape parameters into a vector

$$\mathbb{R}^6 \ni q = (\delta q_1^1, \delta q_2^1, \delta q_1^2, \delta q_2^2, \delta q_1^3, \delta q_2^3)^\top, \tag{B.10}$$

then we can express the discrete shape derivative via

$$\mathbb{R}^{1\times 1\times 6} \ni (\det J)' = \begin{pmatrix} (q_2^2 - q_2^1) - (q_2^3 - q_2^1) \\ (q_1^3 - q_1^1) - (q_1^2 - q_1^1) \\ (q_2^3 - q_2^1) \\ -(q_1^3 - q_1^1) \\ -(q_2^2 - q_2^1) \\ (q_1^2 - q_1^1) \end{pmatrix}^\top. \tag{B.11}$$

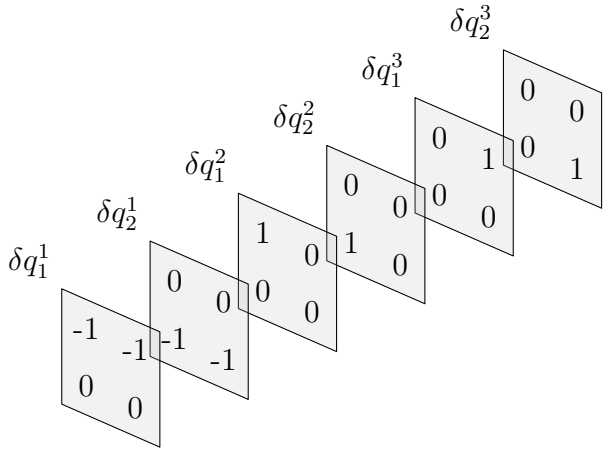

FIGURE 13. The discrete shape gradient $J' \in \mathbb{R}^{2\times 2\times 6}$.

For the matrix $J \in \mathbb{R}^{2\times 2}$, this gets a little harder to visualize. Using the same ordering (B.10), and recalling equation (B.8), we have

$$(J')_{ijk} = \frac{\partial J_{ij}}{\partial q_k}, \tag{B.12}$$

which is depicted in Figure 13. Recalling that we required the shape derivative of J^{-1} rather than of J, we must make use of the identity $JJ^{-1} = I$:

$$J'J^{-1} + J(J^{-1})' = 0,$$

which implies that

$$(J^{-1})' = -J^{-1}J'J^{-1}. \tag{B.13}$$

These Jacobian transformations are applied to local state and adjoint vectors before contraction against J'. It then remains to translate the six components corresponding to the local shape parameters into a global gradient vector.

We can now comment on the relations between the expressions derived in this section and the terms involving $\boldsymbol{V}$ in equation (2.8). If we assume that the perturbation $\boldsymbol{V}$ is piecewise-linear on an element, then analogous to (B.5), we have

$$\hat{\boldsymbol{V}}(\eta) = v^1 + \underbrace{\begin{pmatrix} v^2 - v^1 & v^3 - v^1 \end{pmatrix}}_{\equiv J_v} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}. \tag{B.14}$$

Then, we have

$$\nabla \boldsymbol{V}(x) = \nabla \hat{\boldsymbol{V}}(\eta) \nabla x(\eta)^{-1} = J_v J^{-1}. \tag{B.15}$$

On the other hand, according to (B.13), we can differentiate the term $J^{-1}J^{-\top}$ appearing in equation (B.6) as follows:

$$\begin{aligned} (J^{-1}J^{-\top})' =& (J^{-1})'J^{-\top} + J^{-1}(J^{-\top})' \\ =& - J^{-1}J'J^{-1}J^{-\top} - J^{-1}J^{-\top}(J^{\top})'J^{-\top} \\ =& - J^{-1}\left(J'J^{-1} + J^{-\top}(J^{\top})'\right)J^{-\top}. \end{aligned}$$

Given perturbation directions v^1, v^2, v^3 for the nodes, J' reduces to J_v, so the correspondence between this term and $\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^{\top}$ is established.

Likewise, with div $\boldsymbol{V}$, we have

$$\text{div } \boldsymbol{V} = \text{tr } (\nabla \boldsymbol{V}) = \text{tr } (J_v J^{-1}) \tag{B.16}$$

$$\begin{aligned} =& \frac{1}{\det J}\left((v_1^2 - v_1^1)(x_2^3 - x_2^1) + (v_1^3 - v_1^1)(x_2^1 - x_2^2) + \right. \\ & \left. (v_2^2 - v_2^1)(x_1^1 - x_1^3) + (v_2^3 - v_2^1)(x_1^2 - x_1^1)\right). \end{aligned} \tag{B.17}$$

Associating the perturbations $v^i \sim \delta q^i$, and comparing equations (B.11) and (B.17), we see the equivalence of the two formulations. Thus, we conclude that in this case, discretization of (2.8) and differentiation of (B.6)-(B.7) are equivalent.

References

[1] G. Allaire. *Conception optimale de structures*, volume 58 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, 2007. With the collaboration of Marc Schoenauer (INRIA) in the writing of Chapter 8.

[2] H. Antil, S. S. Hardesty, and M. Heinkenschloss. Supplementary materials: Shape optimization of shell structure acoustics. Technical report, Department of Computational and Applied Mathematics, Rice University, 2016. www.caam.rice.edu/∼heinken/mh_publications.html.

[3] H. Antil, S. S. Hardesty, and M. Heinkenschloss. Shape optimization of shell structure acoustics. *SIAM J. Control Optim.*, 55(3):1347–1376, 2017.
[4] E. Bängtsson, D. Noreland, and M. Berggren. Shape optimization of an acoustic horn. *Computer methods in applied mechanics and engineering*, 192(11-12):1533–1571, 2003.
[5] S. Bartels and G. Wachsmuth. Numerical approximation of optimal convex shapes. *arXiv preprint arXiv:1810.10735*, 2018.
[6] M. Berggren. A unified discrete–continuous sensitivity analysis method for shape optimization. In *Applied and numerical partial differential equations*, pages 25–39. Springer, 2010.
[7] P. Bochev, H. C. Edwards, R. C. Kirby, K. Peterson, and D. Ridzal. Solving PDEs with Intrepid. *Scientific Programming*, 20(2):151–180, 2012.
[8] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries*, volume 22 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2011. Metrics, analysis, differential calculus, and optimization.
[9] P. Gangl. *Sensitivity-Based Topology and Shape Optimization with Application to Electrical Machines*. PhD thesis, Johannes Kepler Universität Linz, 2016.
[10] S. S. Hardesty. *Optimization of shell structure acoustics*. ProQuest LLC, Ann Arbor, MI, 2010. Thesis (Ph.D.)–Rice University.
[11] R. Hiptmair and A. Paganini. Shape optimization by pursuing diffeomorphisms. *Comput. Methods Appl. Math.*, 15(3):291–305, 2015.
[12] R. Hiptmair, A. Paganini, and S. Sargheini. Comparison of approximate shape gradients. *BIT*, 55(2):459–485, 2015.
[13] C. Leithäuser, R. Pinnau, and R. Feßler. Designing polymer spin packs by tailored shape optimization techniques. *Optimization and Engineering*, 19(3):733–764, 2018.
[14] R. Meske, J. Sauter, and E. Schnack. Nonparametric gradient-less shape optimization for real-world applications. *Structural and Multidisciplinary Optimization*, 30(3):201–218, 2005.
[15] S. A Nørgaard, M. Sagebaum, N. R. Gauger, and B. S. Lazarov. Applications of automatic differentiation in topology optimization. *Structural and Multidisciplinary Optimization*, 56(5):1135–1146, 2017.
[16] A. Paganini and R. Hiptmair. Approximate Riesz representatives of shape gradients. In *IFIP Conference on System Modeling and Optimization*, pages 399–409. Springer, 2015.
[17] O. Pironneau. *Optimal shape design for elliptic systems*. Springer Series in Computational Physics. Springer-Verlag, New York, 1984.
[18] S. Schmidt. Weak and strong form shape hessians and their automatic generation. *SIAM Journal on Scientific Computing*, 40(2):C210–C233, 2018.
[19] V. H. Schulz, M. Siebenborn, and K. Welker. Efficient PDE constrained shape optimization based on Steklov–Poincaré-type metrics. *SIAM Journal on Optimization*, 26(4):2800–2819, 2016.
[20] J. Sokołowski and J.-P. Zolésio. *Introduction to shape optimization*, volume 16 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.
[21] K. Sturm. Shape optimization with nonsmooth cost functions: from theory to numerics. *SIAM Journal on Control and Optimization*, 54(6):3319–3346, 2016.

Sean Hardesty, Computational Solid Mechanics and Structural Dynamics, Sandia National Laboratories, Albuquerque, NM 87123, USA.
E-mail address: shardes@sandia.gov

Harbir Antil, The Center for Mathematics and Artificial Intelligence (CMAI) and Department of Mathematical Sciences, George Mason University, Fairfax, VA 22030, USA.
E-mail address: hantil@gmu.edu

Drew P. Kouri, Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM 87123, USA.
E-mail address: dpkouri@sandia.gov

Denis Ridzal, Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM 87123, USA.
E-mail address: dridzal@sandia.gov