COMPARISON OF SHAPE DERIVATIVES USING CUTFEM FOR ILL-POSED BERNOULLI FREE BOUNDARY PROBLEM [*]

ERIK BURMAN [†], CUIYU HE [‡], AND MATS G. LARSON [§]

Abstract. In this paper we study and compare three types of shape derivatives for free boundary identification problems. The problem takes the form of a severely ill-posed Bernoulli problem where only the Dirichlet condition is given on the free (unknown) boundary, whereas both Dirichlet and Neumann conditions are available on the fixed (known) boundary. Our framework resembles the classical shape optimization method in which a shape dependent cost functional is minimized among the set of admissible domains. The position of the domain is defined implicitly by the level set function. The steepest descent method, based on the shape derivative, is applied for the level set evolution. For the numerical computation of the gradient, we apply the Cut Finite Element Method (CutFEM), that circumvents meshing and re-meshing, without loss of accuracy in the approximations of the involving partial differential models.

We consider three different shape derivatives. The first one is the classical shape derivative based on the cost functional with pde constraints defined on the continuous level. The second shape derivative is similar but using a discretized cost functional that allows for the embedding of CutFEM formulations directly in the formulation. Different from the first two methods, the third shape derivative is based on a discrete formulation where perturbations of the domain are built into the variational formulation on the unperturbed domain. This is realized by using the so-called boundary value correction method that was originally introduced to allow for high order approximations to be realized using low order approximation of the domain.

The theoretical discussion is illustrated with a series of numerical examples showing that all three approaches produce similar result on the proposed Bernoulli problem.

Key words. Ill-posed free boundary Bernoulli problem; Cut Finite Element Method; Level set method; non-fitted mesh;

AMS subject classifications. 65N20,65N21,65N30

1. Introduction.

This paper deals with the free boundary identification of the ill-posed free boundary Bernoulli problem. Comparing to the classical free boundary Bernoulli problem, this paper studies the free boundary problems for which only Dirichlet data is given on the free (unknown) boundary and Cauchy data is available on the fixed (known) boundary. Such problems are found for instance in models where perfectly insulated obstacles [1] need to be detected from data. Following [16] we use the cut finite element method (CutFEM) together with a level set approach to numerically identify the free boundary using the shape optimization method. The level set method is highly flexible in handling topology changes and has been widely used for inverse obstacle and optimal design problems [35, 34, 11, 39, 2, 3, 6, 13]. Since the domain of computation changes in each iteration of the shape optimization method, it is advantageous to use a fictitious domain type numerical method, provided a sufficient accuracy can be ensured. This is the rationale for combining the CutFEM with the level set method. The CutFEM additionally features the following advantages: (1)

[*]Submitted to the editors of Journal of Scientific Computing.

Funding: EB and CH were funded by the EPSRC grant EP/P01576X/1. ML was funded by The Swedish Foundation for Strategic Research Grant No. AM13-0029, the Swedish Research Council Grants No. 2017-03911 and the Swedish Research Program Essence

[†]Department of Mathematics, University College London, Gower Street, London, UK–WC1E 6BT, United Kingdom (e.burman@ucl.ac.uk)

[‡]Department of Mathematics, University College London, Gower Street, London, UK–WC1E 6BT, United Kingdom (c.he@ucl.ac.uk)

[§]Department of Mathematics and Mathematical Statistics, Umeå University, SE-90187 Umeå, Sweden (mats.larson@umu.se)

CutFEMs have been designed and analyzed for a large number of PDE models and many types of boundary conditions, (2) for interface problems, CutFEM requires no special construction for basis functions, c.f. the immersed finite element method, the generalized finite element method [40, 37], and (3) optimal accuracy in the bulk and on the boundary can be achieved. The cutFEM method has previously been applied in combination with the level set approach to various shape optimization problems, for instance in [38, 17, 5, 18].

To solve the shape optimization problem, we apply the a steepest descent type algorithm algorithm. The gradient for the shape-dependent cost functional is the so-called shape derivative. The main objective of the present work is to design and compare different types of shape derivatives in the algorithm. Firstly we recall the classical shape derivative that is obtained using the classical shape sensitivity analysis [27] on the continuous level. To obtain the numerical approximation of the shape derivative for the iterative procedure, the solutions in the derivative formulas are replaced directly by their corresponding numerical approximations. We will refer this derivative as the *continuous shape derivative (SD)*. We note here that the shape derivative derived from the continuous level has two equivalent forms by the structure theorem of Hadamard and Zolésio [27, 25], i.e., the domain and boundary representations. Assuming enough regularity on the continuous level those two forms are equivalent. However, the applicability of the domain form is in principle wider, since it requires lower regularity. Moreover, it has been proven to possess certain superconvergence properties compared to the boundary formulation [30, 29, 31]. In this work, we also utilize the domain form.

We note that directly replacing the continuous SD by its numerical approximation only yields an approximate gradient, whose accuracy depends on the mesh-size and that this may prohibit convergence to the minimizer on a fixed mesh. A natural solution is to perform the shape sensitivity analysis directly on the discretized cost functional which allows for the embedding of CutFEM formulations. We will refer this derivative as the *discrete SD*. The resulting advantage for discrete SD is exactness on the mesh-scale considered. Nevertheless, the discrete SD has more complex representation since the discretized cost functional contains significantly more terms than the continuous one. Moreover, the discrete SD in general is not a function in the finite element space and therefore approximation is still inevitable in the final step of the construction of the shape derivative.

For the classical shape sensitivity analysis, the shape derivative is obtained by perturbing the domain and taking the limit for small perturbations. Contrary to such a classical analysis used for the previous SDs, the third shape derivative introduced herein, is defined using only the unperturbed domain. Infinitesimal perturbations of the domain are instead introduced through a boundary correction approach using the weakly imposed boundary conditions that are characteristic of CutFEM. Boundary correction method is a technique to create high order finite element approximations for domains with smooth boundary when using a low order approximation of the domain. Optimal order estimates are obtained through an extrapolation procedure on the boundary [10, 20, 32, 23, 21, 4]. This type shape derivative is also exact as it is based on the discretized functional. We remark that such shape derivative enjoys a much simpler representation that only depends on the boundary terms in the Nitsche, or Lagrange multiplier formulation. This technique, therefore, has great potential to tackle more sophisticated problems where the classical shape derivative is difficult to find. We will refer this derivative as the *boundary SD*. The rigorous justification of this boundary value correction shape derivative will be left for future work, instead we will compare its performance numerically with the two other approaches.

To verify and compare the performance of the three different types of shape derivatives, several numerical experiments are presented in section 6. Since the main objective was to compare the shape derivatives, we only consider a simple steepest descent algorithm for the optimization algorithm and it is expected that convergence can be enhanced by applying a more sophisticated method such as the Levenberg-Marquard method proposed in [12]. The results show that all three shape derivatives have similar performance.

For another level set based identification method not relying on shape derivatives we refer to [8, 9].

The paper is organized as follows. In section 2, we introduce the model problem. Then we introduce the CutFEM for the numerical approximation of the primal and dual solutions in section 3. The various shape derivatives are introduced in section 4. The final optimization algorithm is provided in section 5. Finally, the results for numerical experiments are presented in section 6.

2. Model problem. Let $\hat{\Omega} \subset \mathbb{R}^2$ be a simply connected fixed domain and $\Gamma_f := \partial\hat{\Omega}$. Let $\mathcal{O}$ be a family of admissible bounded connected domains $\Omega \subset \hat{\Omega}$ with the Lipschitz boundary $\partial\Omega = \Gamma_f \cup \Gamma_\Omega$ where Γ_Ω is the free boundary to be determined (see Figure 1 for an example). For simplicity, we assume there is no intersection between Γ_Ω and Γ_f. We consider the interior type ill-posed free boundary Bernoulli problem,

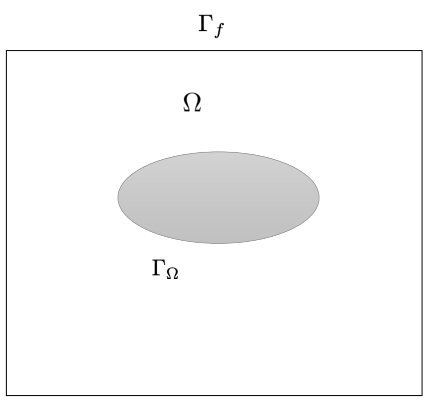


FIG. 1. *The domain Ω with the fixed boundary Γ_f and the free boundary Γ_Ω. Here $\hat{\Omega}$ is the entire square domain.*

i.e., the fixed boundary Γ_f is exterior to Γ_Ω. Find $\Omega^* \in \mathcal{O}$ and $u : \Omega^* \to \mathbb{R}$ such that

$$\begin{aligned} -\triangle u &= f \text{ in } \Omega^*, \\ u &= 0 \text{ on } \Gamma_{\Omega^*}, \\ u &= g_D \text{ on } \Gamma_f, \\ D_n u &= g_N \text{ on } \Gamma_f. \end{aligned} \tag{2.1}$$

The datum (f, g_D, g_N) is chosen such that $f \in L^2(\Omega^*)$, $g_D \in H^{1/2}(\Gamma_f)$ and $g_N \in H^{-1/2}(\Gamma_f)$. $D_n u := \nabla u \cdot \boldsymbol{n}$ where $\boldsymbol{n}$ is the unit outer normal vector to the domain. It is known that, provided the data f, g_D, g_N are compatible with a solution Γ_{Ω^*}, the solution is unique. This follows by a unique continuation argument from the Cauchy data on Γ_f. For a proof in the context of scattering problems we refer to [24, Theorem 2].

To represent the free boundary Γ_Ω, we use the level set method. To be precise, we utilize a level set function $\phi(x)$ for the domain Ω such that

$$\phi(x) \begin{cases} > 0 & \text{if } x \notin \Omega, \\ = 0 & \text{if } x \in \Gamma_\Omega, \\ < 0 & \text{if } x \in \Omega. \end{cases} \tag{2.2}$$

Note that the level set function is not unique and its value away from the free boundary is not critical, provided the gradient of the level set function does not degenerate. A common example for instance is the distance function to the free boundary.

For an arbitrary $\Omega \in \mathcal{O}$, the system (2.1) is over-determined and therefore the solution may not exist. Our goal is to identify the free boundary Γ_{Ω^*} starting from an initial guess Γ_Ω through the shape optimization method. We firstly rephrase the problem (2.1) as a constrained PDE minimization problem.

Define the spaces

$$H^1_{0,\Gamma_\Omega}(\Omega) := \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_\Omega\}, \tag{2.3}$$

$$H^1_0(\Omega) := \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}. \tag{2.4}$$

Recall that $\partial\Omega = \Gamma_\Omega \cup \Gamma_f$. Let $(\cdot,\cdot)_\Omega$ denote the L^2-scalar product over $\Omega \subset \mathbb{R}^2$ and $\langle \cdot,\cdot\rangle_\Gamma$ the L^2-scalar product over the curve $\Gamma \subset \mathbb{R}^2$. The L^2-norm over a subset X of $\mathbb{R}^s$, $s = 1, 2$, will be denoted by $\|\cdot\|_X$.

We now rewrite (2.1) as follows: find $\Omega^* \in \mathcal{O}$ such that

$$J(\Omega^*) = \min_{\Omega\in\mathcal{O}} J(\Omega) \quad \forall\, \Omega \in \mathcal{O}, \tag{2.5}$$

where the cost functional is defined by

$$J(\Omega) = \frac{1}{2} h^{-1} \|g_D - u(\Omega)\|^2_{\Gamma_f}, \tag{2.6}$$

where h is a constant that will be chosen as the mesh size of the finite element mesh introduced later, and $u(\Omega) \in H_{0,\Gamma_\Omega}(\Omega)$ satisfies

$$a(u, v) := (\nabla u, \nabla v)_\Omega = (f, v)_\Omega + \langle g_N, v\rangle_{\Gamma_f} \quad \forall\, v \in H_{0,\Gamma_\Omega}(\Omega). \tag{2.7}$$

The corresponding Lagrangian for the constrained minimization problem (2.5) follows:

$$\mathcal{L}(\Omega, w, v) = \frac{1}{2} h^{-1} \|g_D - w\|^2_{\Gamma_f} - a(w, v) + l(v) \tag{2.8}$$

where $l(v) = (f, v)_\Omega + \langle g_N, v\rangle_{\Gamma_f}$.

The critical point of (2.8), denoted by $(u(\Omega), p(\Omega))$, is obtained through taking the Fréchet derivative with respect to (w, v). This leads to the solution of a decoupled primal and adjoint equation. For the primal variable $u(\Omega)$, we solve (2.7). In strong form, we note that (2.7) corresponds to the following well-posed forward problem:

$$\begin{aligned} -\triangle u(\Omega) &= f \text{ in } \Omega, \\ u(\Omega) &= 0 \text{ on } \Gamma_\Omega, \\ D_n u(\Omega) &= g_N \text{ on } \Gamma_f. \end{aligned} \tag{2.9}$$

For the adjoint solution $p(\Omega)$, we obtain the following weak formulation: find $p(\Omega) \in H^1_{0,\Gamma_\Omega}(\Omega)$ such that

$$(\nabla w, \nabla p(\Omega))_\Omega = h^{-1} \langle u - g_D, w \rangle_{\Gamma_f} \quad \forall w \in H^1_{0,\Gamma_\Omega}(\Omega). \tag{2.10}$$

When there is low risk of ambiguity, we replace $(u(\Omega), p(\Omega))$ by (u, p).

Remark 2.1. If $\Omega = \Omega^*$ we have $u = g_D$ on Γ_f and hence $p \equiv 0$ in Ω^*.

Remark 2.2. The relation between (2.1) and (2.5) is as follows. If Ω^* is the solution to (2.1) then it is also the solution to (2.5). The inverse is also true, by the uniqueness of the inclusion, however there may be local minima that complicate the identification.

Below we present the algorithm of shape optimization using gradient descent iteration to solve (2.5). For simplicity, we restrict our discussion only in the two dimensional case. However, the algorithm and the related analysis can be directly extended to three dimensions.

Algorithm 2.1 Steepest Descent Shape Optimization Method.

Choose an initial level set $\phi(x, 0)$ and set $\Omega = \{x \in \hat{\Omega}, \phi(x, 0) \leq 0\}$ and $\Gamma_\Omega = \{x \in \hat{\Omega}, \phi(x, 0) = 0\}$.
Iterate until the stopping criteria is satisfied:

– Compute the primal and dual solutions $u(\Omega)$ and $p(\Omega)$ for (2.7) and (2.10), respectively.
– Compute the shape derivative

$$\boldsymbol{\beta} := \operatorname*{argmin}_{\boldsymbol{\theta} \in U_{ad}} D_{\Omega,\boldsymbol{\theta}} \mathcal{L}(\Omega, u, p)$$

where $D_{\Omega,\boldsymbol{\theta}} \mathcal{L}(\Omega, u, p)$ is the shape derivative of $\mathcal{L}$ in the direction $\boldsymbol{\theta}$ and U_{ad} is the admissible set for $\boldsymbol{\theta}$.
– Compute $\phi(x, \tau)$ by solving a transport equation in the direction $\boldsymbol{\beta}$ on $\Omega \times [0, \tau(\mathcal{L}, \boldsymbol{\beta})]$.
– Update $\phi(x, 0) = \phi(x, T)$ and set $\Omega = \{x \in \hat{\Omega}, \phi(x, 0) \leq 0\}$, $\Gamma_\Omega = \{x \in \hat{\Omega}, \phi(x, 0) = 0\}$.

3. Approximation of primal and dual solutions using CutFEM. In this section we approximate the primal and dual solution for (2.7) and (2.10), respectively, using the CutFEM method. The main advantages of the CutFEM is that no meshing or re-meshing procedure is needed to fit the moving boundary. The background domain $\hat{\Omega}$, for simplicity, is assumed to be a regular domain, e.g., a unit square. Moreover, stability and accuracy of CutFEM, similar to standard FEM, are guaranteed both in the bulk and on the boundary given proper stabilization.

Let $\mathcal{T} = \{K\}$ be a shape regular triangular partition of $\hat{\Omega}$ and $h = \max_{K \in \mathcal{T}} h_K$ where h_K is the diameter of K. Aldo denote by $\boldsymbol{n}_K$ the outer normal unit vector to K. Define the active computational domain $\Omega_h = \cup\{K \in \mathcal{T}, K \cap \Omega \neq \emptyset\}$, and the space on Ω_h

$$V_h(\Omega_h) = \{v \in H_1(\Omega_h) : v|_K \in P_1(K)\ \forall K \subset \Omega_h\},$$

and, for $v, w \in V_h(\Omega_h)$, define the bilinear form

$$a_h(w, v) := \tilde{a}_h(w, v) + j(w, v) \tag{3.1}$$

with

$$\tilde{a}_h(w,v) = (\nabla w, \nabla v)_\Omega - \langle D_n w, v\rangle_{\Gamma_\Omega} - \langle D_n v, w\rangle_{\Gamma_\Omega} + \beta h^{-1} \langle w, v\rangle_{\Gamma_\Omega}, \tag{3.2}$$

and

$$j(w,v) = \sum_{F\in\mathcal{E}_I(\Omega_h)} \gamma h \int_F [\![D_n w]\!][\![D_n v]\!]\, ds, \tag{3.3}$$

where $\mathcal{E}_I(\Omega_h) = \{F = K_1 \cap K_2, K_1, K_2 \subset \Omega_h\}$ denotes the set of all interior edges in the active computational domain Ω_h. The form $j(w,v)$ is the so-called ghost penalty stabilization [14] and $[\![D_n v]\!]|_F := (\nabla v|_K \cdot \boldsymbol{n}_K) + (\nabla v|_{K'} \cdot \boldsymbol{n}_{K'})$ for $F = K \cap K'$, which is the normal flux jump on F. Note that we added the ghost penalty stabilization for all the interior edges in Ω_h. Nevertheless, the stabilization may be localized to the interior edges close to the interface zone without affecting the accuracy of the method.

Considering the following variational problems: find $u_h \in V_h(\Omega_h)$ such that

$$a_h(u_h, v) = (f, v)_\Omega + \langle g_N, v\rangle_{\Gamma_f} \quad \forall v \in V_h(\Omega_h), \tag{3.4}$$

find $p_h \in V_h(\Omega_h)$ such that

$$a_h(w, p_h) = h^{-1} \langle u_h - g_D, w\rangle_{\Gamma_f} \quad \forall\, w \in V_h(\Omega_h). \tag{3.5}$$

Remark 3.1. Note that in the above formulations all Dirichlet boundary conditions on Γ_Ω are imposed weakly using Nitsche's method [33].

4. Shape derivatives.

In this section, our goal is to derive the formulas for different types of shape derivatives. We firstly discuss some basic definitions and provide some existing results.

4.1. Definition of the shape derivative.

For $\Omega \in \mathcal{O}$, we let $W(\Omega, \mathbb{R}^2)$ denotes the space of sufficiently smooth vector fields $\boldsymbol{\theta} : \Omega \to \mathbb{R}^2$ such that $\boldsymbol{\theta} \equiv 0$ on Γ_f. For a vector field $\boldsymbol{\theta} \in W(\Omega, \mathbb{R}^2)$, we define the map

$$T_{t,\boldsymbol{\theta}} : x \in \Omega \to x + t\boldsymbol{\theta}(x) \in \Omega_t(\boldsymbol{\theta}) \subset \mathbb{R}^2. \tag{4.1}$$

The variable t is interpreted as a pseudo-time. For small t the mapping $\Omega \to \Omega_t(\boldsymbol{\theta})$ is assumed to be a bijection. We also assume that $\Omega_t(\boldsymbol{\theta}) \in \mathcal{O}$ for any $t \in I = \{-\delta, \delta\}$, with $\delta > 0$ small enough. When there is no risk of confusion, we let $\Omega_t = \Omega_t(\boldsymbol{\theta})$ and $T_t = T_{t,\boldsymbol{\theta}}$.

The shape derivative of the cost functional $\mathcal{L}(\Omega, u(\Omega), p(\Omega))$ in the direction of $\boldsymbol{\theta}$ is defined as

$$D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega, u(\Omega), p(\Omega)) := \lim_{t\to 0} \frac{1}{t}(\mathcal{L}(\Omega_t(\boldsymbol{\theta}), u(\Omega_t(\boldsymbol{\theta})), p(\Omega_t(\boldsymbol{\theta}))) - \mathcal{L}(\Omega, u(\Omega), p(\Omega))). \tag{4.2}$$

For a scalar function $v(x,t) : \Omega \times I \to \mathbb{R}$ that is smooth enough, we define its material derivative in the direction $\boldsymbol{\theta}$ by

$$D_{t,\boldsymbol{\theta}} v(x) = \lim_{t\to 0} \frac{v(x(t),t) - v(x(0),0)}{t} \tag{4.3}$$

where $x(t) = T_{t,\boldsymbol{\theta}}(x) = x + t\boldsymbol{\theta}(x)$ and $x(0) = x$. We also define the pseudo-time derivative by

$$\partial_t v(x) = \lim_{t\to 0} \frac{v(x,t) - v(x,0)}{t}. \tag{4.4}$$

By the chain rule it is easy to see that

$$D_{t,\boldsymbol{\theta}}\, v = \partial_t v + \boldsymbol{\theta} \cdot \nabla v. \tag{4.5}$$

The product rule holds for the material derivative:

$$D_{t,\boldsymbol{\theta}}\, (vw) = w D_{t,\boldsymbol{\theta}} v + v D_{t,\boldsymbol{\theta}}\, w. \tag{4.6}$$

For easier representation, we replace the notations by $\dot{v} := D_{t,\boldsymbol{\theta}} v$ and $v' := \partial_t v$ when there is no risk of ambiguity.

4.2. Shape derivatives of linear and bilinear forms. We now state several technical results that allow us to derive the explicit representation of the shape derivative acting on the cost functional. The shape derivatives associated to the bulk terms are fairly standard and the proofs of these results follow the ideas of [36, 25]. For the cutFEM method however, we also need shape derivatives of integral forms over the boundaries and of stabilization terms. All proofs are reported in the appendix for completeness. The following concise notation for the symmetric gradient of the deformation vector field $\boldsymbol{\theta}$ will be used below, $S(\boldsymbol{\theta}) = \nabla\boldsymbol{\theta} + (\nabla\boldsymbol{\theta})^t$ and on the interface Γ_Ω we define $\nabla_\Gamma \cdot \boldsymbol{\theta} = \nabla \cdot \boldsymbol{\theta} - (\nabla\boldsymbol{\theta} \cdot \boldsymbol{n}) \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ is the outer normal vector of Γ.

LEMMA 4.1. *Let Ω be an open set in $\mathbb{R}^2$, $\Gamma_\Omega \subset \partial\Omega$ is a closed curve, and $\boldsymbol{\theta} : \mathbb{R}^2 \to \mathbb{R}^2$ be an injective differentiable mapping. Then the following equalities hold:*

$$\begin{aligned} D_{\Omega,\boldsymbol{\theta}} \int_\Omega \phi \, dx &= \int_\Omega (\dot{\phi} + (\nabla \cdot \boldsymbol{\theta})\phi) \, dx, \\ D_{\Omega,\boldsymbol{\theta}} \int_{\Gamma_\Omega} \psi \, ds &= \int_{\Gamma_\Omega} (\dot{\psi} + (\nabla_\Gamma \cdot \boldsymbol{\theta})\psi) \, ds, \end{aligned} \tag{4.7}$$

where we assume that $\phi(x,t), \psi(x,t) : \mathbb{R}^2 \times I \to \mathbb{R}$ are functions smooth enough for the expressions of (4.7) to be well defined.

LEMMA 4.2. *With the same assumptions for Ω and $\boldsymbol{\theta}$ as in Lemma 4.1, the following relation holds:*

$$\begin{aligned} D_{\Omega,\boldsymbol{\theta}} \int_\Omega \nabla w \cdot \nabla v \, dx = &\int_\Omega (\nabla \cdot \boldsymbol{\theta})(\nabla w \cdot \nabla v) - (S(\boldsymbol{\theta}) \cdot \nabla w) \cdot \nabla v \, dx \\ &+ \int_\Omega \nabla \dot{w} \cdot \nabla v + \nabla \dot{v} \cdot \nabla w \, dx, \end{aligned} \tag{4.8}$$

where we assume that $w(x,t), v(x,t) : \mathbb{R} \times I \to \mathbb{R}$ are functions smooth enough for (4.8) to be well defined.

LEMMA 4.3. *With the same assumptions for Ω and $\boldsymbol{\theta}$ as in Lemma 4.1, the following relation holds:*

$$\begin{aligned} D_{\Omega,\boldsymbol{\theta}} \int_{\Gamma_\Omega} (D_n w) v \, ds = &\int_{\Gamma_\Omega} (\nabla \cdot \boldsymbol{\theta})(D_n w) v - (S(\boldsymbol{\theta}) \cdot \nabla w) \cdot \boldsymbol{n} v \, ds \\ &+ \int_{\Gamma_\Omega} (D_n \dot{w}) v \, ds + (D_n w) \dot{v} \, ds, \end{aligned} \tag{4.9}$$

where we assume that $w(x,t), v(x,t) : \mathbb{R} \times I \to \mathbb{R}$ are functions smooth enough for (4.9) to be well defined.

In the following Lemma we provide the shape derivative for the ghost penalty stabilization term.

LEMMA 4.4. *Assume that $w(x,t), v(x,t) \in H^1(\Omega,t)$ and that locally on each triangle K, $w(x,t)|_K$, $v(x,t)|_K \in H^{3/2+\epsilon}(K)$ for some $\epsilon > 0$. Then there holds for each $F \in \mathcal{E}_I(\Omega_h)$*

$$D_{\Omega,\boldsymbol{\theta}} \int_F [\![D_n w]\!][\![D_n v]\!]\, ds = \int_F ([\![D_n \dot{w}]\!][\![D_n v]\!] + [\![D_n w]\!][\![D_n \dot{v}]\!])\, ds + \Upsilon_F(w,v) \tag{4.10}$$

where

$$\begin{aligned}\Upsilon_F(w,v) = &\int_F [\![(\nabla\cdot\boldsymbol{\theta}) D_n w - (S(\boldsymbol{\theta})\cdot\nabla w)\cdot\boldsymbol{n}]\!][\![D_n v]\!]\, ds \\ &+ \int_F [\![(\nabla\cdot\boldsymbol{\theta}) D_n v - (S(\boldsymbol{\theta})\cdot\nabla v)\cdot\boldsymbol{n}]\!][\![D_n w]\!]\, ds \\ &- \int_F [\![D_n w]\!][\![D_n v]\!]\, \nabla_F\cdot\boldsymbol{\theta}\, ds.\end{aligned} \tag{4.11}$$

4.3. Continuous SD. In this subsection, we obtain the continuous SD of the cost functional $\mathcal{L}(\Omega,u,p)$ in the direction $\boldsymbol{\theta}$. From this point, we assume the admissible set for $\boldsymbol{\theta}$ is $H^1(\hat{\Omega})^d$. In the numerical approximation, we will simply replace the continuous solutions by their corresponding numerical approximations. Note that continuous SD is independent of the numerical method, and, therefore, the shape derivative is not exact. The error in the gradient will be of optimal order asymptotically, if the CutFEM solution has optimal error estimates in $W^{1,4}(\Omega)$ and $L^4(\Omega)$, see [16].

On $\Omega_t(\boldsymbol{\theta})$, $t \in [-\delta,\delta]$, $u(x,t) \in H^1_{0,\Gamma_{\Omega_t}}(\Omega_t)$ and $p(x,t) \in H^1_{0,\Gamma_{\Omega_t}}(\Omega_t)$ are defined such that

$$(\nabla u(x,t), \nabla v)_{\Omega_t} = (f,v)_{\Omega_t} + \langle g_N, v\rangle_{\Gamma_f} \quad \forall\, v \in H^1_{0,\Gamma_{\Omega_t}}(\Omega_t) \tag{4.12}$$

and

$$(\nabla v, \nabla p(x,t))_{\Omega_t} = h^{-1}\langle u(x,t) - g_D, v\rangle_{\Gamma_f} \quad \forall\, v \in H^1_{0,\Gamma_{\Omega_t}}(\Omega_t). \tag{4.13}$$

Immediately we have that $\dot{p} = \dot{u} = 0$ on Γ_Ω, therefore $\dot{u} \in H^1_{0,\Gamma_\Omega}(\Omega)$ and $\dot{p} \in H^1_{0,\Gamma_\Omega}(\Omega)$.

LEMMA 4.5. *Let $\mathcal{L}(\Omega,u,p)$ be defined in (2.8). Then its shape derivative in the direction $\boldsymbol{\theta}$ has the following representation:*

$$\begin{aligned}&D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega,u,p) \\ &\quad = \int_\Omega (\nabla\cdot\boldsymbol{\theta})\,(fp - \nabla u\cdot\nabla p)\; dx + \int_\Omega (S(\boldsymbol{\theta})\cdot\nabla u)\cdot\nabla p\, dx + \int_\Omega (\nabla f\cdot\boldsymbol{\theta}) p\, dx.\end{aligned} \tag{4.14}$$

Proof. Rearrange $\mathcal{L}(\Omega,u,p)$ such that

$$\mathcal{L}(\Omega,u,p) \triangleq \mathcal{A}_1 + \mathcal{A}_2 \tag{4.15}$$

where

$$\mathcal{A}_1 = -(\nabla u,\nabla p)_\Omega + (f,p)_\Omega\,, \quad \mathcal{A}_2 = \frac{1}{2}h^{-1}\langle g_D - u, g_D - u\rangle_{\Gamma_f} + \langle g_N, p\rangle_{\Gamma_f}\,.$$

Note that $\dot{f} = \nabla f \cdot \boldsymbol{\theta}$ since $f' = 0$. By Lemma 4.1 and Lemma 4.2, we then have

(4.16)
$$
\begin{aligned}
D_{\boldsymbol{\theta},\Omega}\mathcal{A}_1 &= \int_\Omega (\nabla\cdot\boldsymbol{\theta})(fp - \nabla u\cdot\nabla p)\,dx + \int_\Omega (S(\boldsymbol{\theta})\cdot\nabla u)\cdot\nabla p\,dx \\
&\quad - (\nabla\dot{u},\nabla p)_\Omega - (\nabla u,\nabla\dot{p})_\Omega + (\dot{f},p)_\Omega + (f,\dot{p})_\Omega \\
&= \int_\Omega (\nabla\cdot\boldsymbol{\theta})(fp - \nabla u\cdot\nabla p)\,dx + \int_\Omega (S(\boldsymbol{\theta})\cdot\nabla u)\cdot\nabla p\,dx + \int_\Omega (\nabla f\cdot\boldsymbol{\theta})p\,dx \\
&\quad - (\nabla\dot{u},\nabla p)_\Omega - (\nabla u,\nabla\dot{p})_\Omega + (f,\dot{p})_\Omega.
\end{aligned}
$$

Thanks to the fact that $\dot{u} \in H^1_{0,\Gamma_\Omega}(\Omega)$ and $\dot{p} \in H^1_{0,\Gamma_\Omega}(\Omega)$, together with (2.7) and (2.10), we have

$$
\begin{aligned}
-(\nabla\dot{u},\nabla p)_\Omega - (\nabla u,\nabla\dot{p})_\Omega + (f,\dot{p})_\Omega &= -h^{-1}\left\langle u - g_D, \dot{u}\right\rangle_{\Gamma_f} - \left\langle g_N, \dot{p}\right\rangle_{\Gamma_f} \\
&= -h^{-1}\left\langle u - g_D, u'\right\rangle_{\Gamma_f} - \left\langle g_N, p'\right\rangle_{\Gamma_f}.
\end{aligned} \tag{4.17}
$$

Note that on Γ_f, we have used the fact that $\dot{u} = u'$ and $\dot{p} = p'$, since $\boldsymbol{\theta} = 0$ on Γ_f. By the product and chain rule we immediately have

$$
D_{\boldsymbol{\theta},\Omega}\mathcal{A}_2 = h^{-1}\left\langle u - g_D, u'\right\rangle_{\Gamma_f} + \left\langle g_N, p'\right\rangle_{\Gamma_f}. \tag{4.18}
$$

Combining (4.15)–(4.18) gives (4.14). This completes the proof of the lemma. ∎

4.4. Discrete SD. In this subsection, we obtain the discrete SD of a discrete cost functional $\mathcal{L}_h(\Omega, u_h, p_h)$ in the direction $\boldsymbol{\theta}$ where $\mathcal{L}_h$ is the discrete Lagrangian functional that embeds the CutFEM formulation and (u_h, p_h) are the numerical approximations. As a consequence the shape derivative in the direction $\boldsymbol{\theta}$ is exact for the mesh-scale.

Starting from the Lagrangian (2.8) we define the discrete Lagrangian functional as follows:

$$
\mathcal{L}_h(\Omega, w_h, v_h) = \frac{1}{2}h^{-1}\|g_D - w_h\|^2_{\Gamma_f} - a_h(w_h, v_h) + l(v_h), \tag{4.19}
$$

where a_h is defined in defined in (3.1) and, we recall, $l(v) := (f,v)_\Omega + \left\langle g_N, v\right\rangle_{\Gamma_f}$.

Note that taking the Fréchet derivative with respect to v_h and w_h in (4.19) gives the CutFEM formulation for the critical point (u_h, p_h) that satisfies (3.4) and (3.5), respectively.

To define the discrete SD, firstly we need to define the finite dimensional function space for the perturbed solutions $(u_h(x,t), p_h(x,t))$ on Ω_t. We do this by using a pullback map to Ω where the finite element mesh is triangular and use the standard definition of the finite element space on the reference domain.

For each $K \in \mathcal{T}$, let $K^t = T_{t,\boldsymbol{\theta}}K$. Note that K^t does not necessarily remain as a triangle, however, should be non-degenerate, and its shape is determined by $\boldsymbol{\theta}$. It should be interpreted as an auxiliary perturbed element that only serves in the analysis. Here we further assume that $\boldsymbol{\theta} \in [C^1(\Omega)]^d$. Then, by the inverse function theorem, T_t is a bijection for sufficiently small t and its derivatives are point wise well defined. We also define $\mathcal{T}^t := \{K^t, K \in \mathcal{T}\}$, $\Omega_{h,t} = T_t(\Omega_h)$, and the finite dimensional space on $\Omega_{h,t}$

$$
V_h^t(\Omega_{h,t}) := \{v \in H^1(\Omega_{h,t}), v|_{K^t} \in V_h^t(K^t)\}
$$

where $V_h^t(K^t)$ is defined as $V_h^t(K^t) = V_h(K) \circ T_t^{-1}$. Here $V_h(K) := V_h(\mathcal{T})|_K = P_1(K)$. It is then easy to verify that

$$v_h^t \circ T_t \in V_h(\Omega_h) \quad \forall v_h^t \in V_h^t(\Omega_{h,t}).$$

We now define $u_h(x,t)$ and $p_h(x,t)$ on Ω_t. Let $u_h(x,t)$ and $p_h(x,t)$ be the solution of (3.4) and (3.5) in the space $V_h^t(\Omega_{h,t})$ with integrals on Ω and Γ_Ω replaced by Ω_t and Γ_{Ω_t}, respectively.

LEMMA 4.6. *Let $u_h(x,t)$ and $p_h(x,t)$ be defined as above. Then*

$$\dot{u}_h \in V_h(\Omega_h) \quad \textit{and} \quad \dot{p}_h \in V_h(\Omega_h). \tag{4.20}$$

Proof. By the definition, we have that

$$\begin{aligned} \dot{u}_h(x) &= \lim_{t\to 0} \frac{1}{t}(u_h(x(t),t) - u_h(x,0)) \\ &= \lim_{t\to 0} \frac{1}{t}(u_h(T_t(x),t) - u_h(x,0)). \end{aligned} \tag{4.21}$$

Since both $u_h(T_t(x),t) \in V_h(\Omega_h)$ and $u_h(x,0) \in V_h(\Omega_h)$, we have that $\dot{u}_h \in V_h(\Omega_h)$. The result for $\dot{p}_h$ also holds by the same argument. ∎

In the following lemma we derive the discrete derivative for $\mathcal{L}_h(\Omega, u_h, p_h)$ in the direction $\boldsymbol{\theta}$.

LEMMA 4.7. *Let $\mathcal{L}_h(\Omega, u_h, p_h)$ be defined in (4.19). Then its shape derivative has the following representation in the direction $\boldsymbol{\theta}$:*

$$\begin{aligned} &D_{\Omega,\boldsymbol{\theta}}\mathcal{L}_h(\Omega, u_h(\Omega), p_h(\Omega)) \\ &= \int_\Omega (\nabla\cdot\boldsymbol{\theta})\,(f p_h - \nabla u_h\cdot\nabla p_h)\,dx + \int_\Omega (S(\boldsymbol{\theta})\cdot\nabla u_h)\cdot\nabla p_h\,dx + \int_\Omega (\nabla f\cdot\boldsymbol{\theta}) p_h\,dx \\ &\quad + \int_{\Gamma_\Omega} (\nabla\cdot\boldsymbol{\theta})(D_n u_h)p_h - (S(\boldsymbol{\theta})\cdot\nabla u_h)\cdot\boldsymbol{n} p_h\,ds \\ &\quad + \int_{\Gamma_\Omega} (\nabla\cdot\boldsymbol{\theta})(D_n p_h)u_h - (S(\boldsymbol{\theta})\cdot\nabla p_h)\cdot\boldsymbol{n} u_h\,ds \\ &\quad - \int_{\Gamma_\Omega} \beta h^{-1}(\nabla_\Gamma\cdot\boldsymbol{\theta})u_h p_h\,ds + \sum_{F\in\mathcal{E}_I(\Omega_h)} \gamma h \Upsilon_F(u_h,p_h) \end{aligned} \tag{4.22}$$

Proof. Rearrange $\mathcal{L}_h(\Omega, u_h, p_h)$ such that

$$\mathcal{L}_h(\Omega, u_h, p_h) \triangleq \sum_{i=1}^4 \mathcal{A}_i \tag{4.23}$$

where

$$\mathcal{A}_1 = -(\nabla u_h, \nabla p_h)_\Omega + (f, p_h)_\Omega\,, \quad \mathcal{A}_2 = \frac{1}{2}h^{-1}\left\langle g_D - u_h, g_D - u_h\right\rangle_{\Gamma_f} + \left\langle g_N, p_h\right\rangle_{\Gamma_f}\,,$$

$$\mathcal{A}_3 = \left\langle D_n u_h, p_h\right\rangle_{\Gamma_\Omega} + \left\langle D_n p_h, u_h\right\rangle_{\Gamma_\Omega} - \beta h^{-1}\left\langle u_h, p_h\right\rangle_{\Gamma_\Omega}\,, \quad \mathcal{A}_4 = -j(u_h,p_h).$$

For the first two terms, we derive its shape derivative similarly as in (4.16) and (4.18):

$$\begin{aligned} D_{\boldsymbol{\theta},\Omega}\mathcal{A}_1 = &\int_\Omega (\nabla\cdot\boldsymbol{\theta})(f p_h - \nabla u_h\cdot\nabla p_h) + (S(\boldsymbol{\theta})\cdot\nabla u_h)\cdot\nabla p_h + (\nabla f\cdot\boldsymbol{\theta})p_h\,dx \\ &- (\nabla\dot{u}_h, \nabla p_h)_\Omega - (\nabla u_h, \nabla\dot{p}_h)_\Omega + (f, \dot{p}_h)_\Omega, \end{aligned} \tag{4.24}$$

and

$$D_{\boldsymbol{\theta},\Omega}\mathcal{A}_2 = h^{-1}\langle u_h - g_D, u_h'\rangle_{\Gamma_f} + \langle g_N, p_h'\rangle_{\Gamma_f}. \tag{4.25}$$

For $\mathcal{A}_3$, by Lemma 4.1 and Lemma 4.3 we have

$$\begin{aligned} D_{\boldsymbol{\theta},\Omega}\mathcal{A}_3 = &\int_{\Gamma_\Omega} (\nabla\cdot\boldsymbol{\theta})(D_n u_h)p_h - (S(\boldsymbol{\theta})\cdot\nabla u_h)\cdot\boldsymbol{n}p_h + (D_n\dot{u}_h)p_h + (D_n u_h)\dot{p}_h\, ds \\ &+ \int_{\Gamma_\Omega} (\nabla\cdot\boldsymbol{\theta})(D_n p_h)u_h - (S(\boldsymbol{\theta})\cdot\nabla p_h)\cdot\boldsymbol{n}u_h + (D_n\dot{p}_h)u_h + (D_n p_h)\dot{u}_h\, ds \\ &- \beta h^{-1}\int_{\Gamma_\Omega} (\nabla_\Gamma\cdot\boldsymbol{\theta})u_h p_h + \dot{u}_h p_h + u_h\dot{p}_h\, ds. \end{aligned} \tag{4.26}$$

For $\mathcal{A}_4$, by Lemma 4.4 we have

$$D_{\boldsymbol{\theta},\Omega}\mathcal{A}_4 = -j(u_h,\dot{p}_h) - j(\dot{u}_h,p_h) - \sum_{F\in\mathcal{E}_I(\Omega_h)}\gamma\Upsilon_F(u_h,p_h). \tag{4.27}$$

Thanks to the fact that $\dot{u}_h \in V_h(\Omega_h)$ and $\dot{p}_h \in V_h(\Omega_h)$, with v replaced by $\dot{p}_h$ in (3.4) and w replaced by $\dot{u}_h$ in (3.5), we have

$$\begin{aligned} 0 = &-(\nabla\dot{u}_h,\nabla p_h)_\Omega - (\nabla u_h,\nabla\dot{p}_h)_\Omega + (f,\dot{p}_h)_\Omega \\ &- h^{-1}\langle g_D - u_h,\dot{u}_h\rangle_{\Gamma_f} + \langle g_N,\dot{p}_h\rangle_{\Gamma_f} \\ &+ \langle D_n\dot{u}_h,p_h\rangle_{\Gamma_\Omega} + \langle D_n u_h,\dot{p}_h\rangle_{\Gamma_\Omega} + \langle D_n\dot{p}_h,u_h\rangle_{\Gamma_\Omega} + \langle D_n p_h,\dot{u}_h\rangle_{\Gamma_\Omega} \\ &- \beta h^{-1}\langle\dot{u}_h,p_h\rangle_{\Gamma_\Omega} - \beta h^{-1}\langle u_h,\dot{p}_h\rangle_{\Gamma_\Omega} \\ &- j(u_h,\dot{p}_h) - j(\dot{u}_h,p_h). \end{aligned} \tag{4.28}$$

Combing (4.23)–(4.28) gives (4.22). ∎

Remark 4.8. The directional discrete SD is exact, however, due to the extra terms in the CutFEM formulation it has a more complex representation.

4.5. CutFEM with boundary value correction. In the classical shape sensitivity analysis as utilized for the continuous and discrete SD, the function $u(x,t)$ and $p(x,t)$ are defined on the domain of Ω_t. In this subsection the effect of domain perturbation is included through the boundary correction approach. This means that the perturbed solutions $(u(x,t),p(x,t))$ remain defined on the unperturbed domain Ω for all t, but the effect of domain is included through an extrapolation procedure in the weakly imposed boundary conditions. The idea of the boundary correction approach where weakly imposed boundary conditions are perturbed in order to improve geometry approximation was first introduced in [10]. The extension to CutFEM was considered in [20]. For a recent discussion of the method interpreted as a singular Robin condition we refer to [26]. The idea of extrapolation on the boundary has already been used in the context of the standard Bernoulli problem, see [7]. However the use of boundary value correction as a vehicle for shape sensitivity analysis appears to be new.

Drawing on the ideas on boundary correction for the CutFEM method [20], we modify the weak formulation on the free boundary as follows:

$$\tilde{a}_h^t(w,v) = (\nabla w,\nabla v)_\Omega - \langle D_n w,v\rangle_{\Gamma_\Omega} - \langle D_n v,w\circ T_t\rangle_{\Gamma_\Omega} + \beta h^{-1}\langle w\circ T_t,v\circ T_t\rangle_{\Gamma_\Omega}, \tag{4.29}$$

and

$$a_h^t(w,v) := \tilde{a}_h^t(w,v) + j(w,v).$$

We emphasize that the above modified bilinear form $\tilde{a}_h^t$ is similar to (3.2) but with the Dirichlet condition now imposed on $T_t(\Gamma_\Omega) = \Gamma_{\Omega_t}$ through an extrapolation.

Now, considering the following variational problems: finding $u_h(x,t) \in V_h(\Omega_h)$ such that

$$a_h^t(u_h(x,t), v) = (f,v)_\Omega + \langle g_N, v\rangle_{\Gamma_f} \quad \forall v \in V_h(\Omega_h), \tag{4.30}$$

and finding $p_h(x,t) \in V_h(\Omega_h)$ such that

$$a_h^t(w, p_h(x,t)) = h^{-1}\langle u_h(x,t) - g_D, w\rangle_{\Gamma_f} \quad \forall\, w \in V_h(\Omega_h). \tag{4.31}$$

Note that the above weak formulation is consistent with the following in the strong form:

$$-\triangle u = f \in \Omega, \quad D_n u = g_N \text{ on } \Gamma_f, \quad \text{and} \quad u = 0 \text{ on } \Gamma_{\Omega_t}.$$

We now modify the discrete Lagrangian at pseudo-time t with respect to $\boldsymbol{\theta}$ as follows:

$$\begin{aligned} &\tilde{\mathcal{L}}_h(\Omega_t, u_h(x,t), p_h(x,t)) \\ &= \frac{1}{2}h^{-1}\|g_D - u_h(x,t)\|^2_{\Gamma_f} - a_h^t(u_h(x,t), p_h(x,t)) + l(p_h(x,t)). \end{aligned} \tag{4.32}$$

where $u_h(x,t)$ and $p_h(x,t)$ are the solutions to (4.30) and (4.31), respectively.

Remark 4.9. It is easy to see that

$$\lim_{t\to 0} \tilde{\mathcal{L}}_h(\Omega_t, u_h(t), p_h(t)) = \mathcal{L}_h(\Omega, u_h, p_h).$$

Finally, we define the modified directional shape derivative in the direction $\boldsymbol{\theta}$ by

$$D_{\Omega,\boldsymbol{\theta}}\tilde{\mathcal{L}}_h(\Omega, u_h, p_h) = \lim_{t\to 0}\frac{1}{t}\left(\tilde{\mathcal{L}}_h(\Omega_t, u_h(t), p_h(t)) - \mathcal{L}_h(\Omega, u_h, p_h)\right), \tag{4.33}$$

where u_h, p_h are the solutions on Ω for (3.4) and (3.5), respectively.

4.6. Boundary SD. In this subsection we derive the explicit formula for the Boundary SD defined in (4.33) in the direction $\boldsymbol{\theta}$.

LEMMA 4.10. *Let u_h and p_h be the solutions of (3.4) and (3.5), respectively. We have the following expression for the modified shape derivative defined in (4.33):*

$$D_{\Omega,\boldsymbol{\theta}}\tilde{\mathcal{L}}_h(\Omega, u_h, p_h) = \langle D_n p_h, \nabla u_h\cdot\boldsymbol{\theta}\rangle_{\Gamma_\Omega} - \beta h^{-1}\left(\langle \nabla u_h\cdot\boldsymbol{\theta}, p_h\rangle_{\Gamma_\Omega} + \langle \nabla p_h\cdot\boldsymbol{\theta}, u_h\rangle_{\Gamma_\Omega}\right). \tag{4.34}$$

Proof. By definition we have

$$\begin{aligned} D_{\Omega,\boldsymbol{\theta}}\tilde{\mathcal{L}}_h(\Omega, u_h, p_h) &= \lim_{t\to 0}\frac{1}{t}\Big(\tilde{\mathcal{L}}_h(\Omega_t, u_h(t), p_h(t) - \mathcal{L}_h(\Omega, u_h, p_h)\Big) \\ &= \lim_{t\to 0}\frac{1}{2t}h^{-1}\left(\|u_h(t) - g_D\|^2_{\Gamma_f} - \|u_h - g_D\|^2_{\Gamma_f}\right) \\ &\quad - \lim_{t\to 0}\frac{1}{t}\left(a_h^t(u_h(t), p_h(t)) - a_h(u_h, p_h)\right) \\ &\quad + \lim_{t\to 0}\frac{1}{t}(f, p_h(t) - p_h)_\Omega + \lim_{t\to 0}\frac{1}{t}\langle g_N, p_h(t) - p_h\rangle_{\Gamma_f} \\ &\quad - \lim_{t\to 0}\frac{1}{t}(j(u_h(t), p_h(t)) - j(u_h, p_h)) \\ &\triangleq \sum_{i=1}^{5}\mathcal{A}_i. \end{aligned} \tag{4.35}$$

By direct calculations, we have

$$\mathcal{A}_1 = h^{-1}\left\langle u_h - g_D, u_h' \right\rangle_{\Gamma_f}, \quad \mathcal{A}_3 = (f, p_h')_\Omega, \tag{4.36}$$

$$\mathcal{A}_4 = \left\langle g_N, p_h' \right\rangle_{\Gamma_f}, \qquad\qquad \mathcal{A}_5 = -j(u_h', p_h) - j(u_h, p_h'). \tag{4.37}$$

Expanding and regrouping terms in $a_h^t(\cdot)$ and $a_h(\cdot)$ gives

$$\begin{aligned} -\mathcal{A}_2 =& \lim_{t\to 0}\frac{1}{t}\left(a_h^t(u_h(t), p_h(t)) - a_h(u_h, p_h)\right) \\ =& \lim_{t\to 0}\frac{1}{t}\left((\nabla u_h(t), \nabla p_h(t))_\Omega - (\nabla u_h, \nabla p_h)_\Omega\right) \\ & - \lim_{t\to 0}\frac{1}{t}\left(\left\langle D_n u_h(t), p_h(t)\right\rangle_{\Gamma_\Omega} - \left\langle D_n u_h, p_h\right\rangle_{\Gamma_\Omega}\right) \\ & - \lim_{t\to 0}\frac{1}{t}\left(\left\langle D_n p_h(t), u_h(t)\circ T_t\right\rangle_{\Gamma_\Omega} - \left\langle D_n p_h, u_h\right\rangle_{\Gamma_\Omega}\right) \\ & + \lim_{t\to 0}\frac{1}{t}\beta h^{-1}\left(\left\langle u_h(t)\circ T_t, p_h(t)\circ T_t\right\rangle_{\Gamma_\Omega} - \left\langle u_h, p_h\right\rangle_{\Gamma_\Omega}\right). \end{aligned} \tag{4.38}$$

Applying the product rule, Taylor expansion and neglecting the higher order terms gives

$$\begin{aligned} -\mathcal{A}_2 =& (\nabla u_h', p_h)_\Omega + (\nabla u_h, \nabla p_h')_\Omega - \left\langle D_n u_h', p_h\right\rangle_{\Gamma_\Omega} - \left\langle D_n u_h, p_h'\right\rangle_{\Gamma_\Omega} \\ & - \lim_{t\to 0}\frac{1}{t}\left(\left\langle D_n p_h(t), u_h(t) + t\nabla u_h(t)\cdot\boldsymbol{\theta}\right\rangle_{\Gamma_\Omega} - \left\langle D_n p_h, u_h\right\rangle_{\Gamma_\Omega}\right) \\ & + \lim_{t\to 0}\frac{1}{t}\beta h^{-1}\left(\left\langle u_h(t) + t\nabla u_h(t)\cdot\boldsymbol{\theta}, p_h(t) + t\nabla p_h(t)\cdot\boldsymbol{\theta}\right\rangle_{\Gamma_\Omega} - \left\langle u_h, p_h\right\rangle_{\Gamma_\Omega}\right) \\ =& (\nabla u_h', \nabla p_h)_\Omega + (\nabla u_h, \nabla p_h')_\Omega - \left\langle D_n u_h', p_h\right\rangle_{\Gamma_\Omega} - \left\langle D_n u_h, p_h'\right\rangle_{\Gamma_\Omega} \\ & - \left(\left\langle D_n p_h', u_h\right\rangle_{\Gamma_\Omega} + \left\langle D_n p_h, u_h'\right\rangle_{\Gamma_\Omega} + \left\langle D_n p_h, \nabla u_h\cdot\boldsymbol{\theta}\right\rangle_{\Gamma_\Omega}\right) \\ & + \beta h^{-1}\left(\left\langle u_h', p_h\right\rangle_{\Gamma_\Omega} + \left\langle u_h, p_h'\right\rangle_{\Gamma_\Omega} + \left\langle \nabla u_h\cdot\boldsymbol{\theta}, p_h\right\rangle_{\Gamma_\Omega} + \left\langle u_h, \nabla p_h\cdot\boldsymbol{\theta}\right\rangle_{\Gamma_\Omega}\right). \end{aligned} \tag{4.39}$$

Note that $u_h', p_h' \in V_h(\Omega_h)$. By (3.4) and (3.5) we have

$$\begin{aligned} & (\nabla p_h, \nabla u_h')_\Omega - \left\langle D_n p_h, u_h'\right\rangle_{\Gamma_\Omega} - \left\langle D_n u_h', p_h\right\rangle_{\Gamma_\Omega} + \beta h^{-1}\left\langle p_h, u_h'\right\rangle_{\Gamma_\Omega} + j(p_h, u_h') \\ =& a_h(u_h', p_h) = h^{-1}\left\langle u_h - g_D, u_h'\right\rangle_{\Gamma_f} \end{aligned} \tag{4.40}$$

and

$$\begin{aligned} & (\nabla u_h, \nabla p_h')_\Omega - \left\langle D_n u_h, p_h'\right\rangle_{\Gamma_\Omega} - (D_n p_h', u_h)_{\Gamma_\Omega} + \beta h^{-1}(u_h, p_h')_{\Gamma_\Omega} + j(u_h, p_h') \\ =& a_h(u_h, p_h') = (f, p_h')_\Omega + \left\langle g_N, p_h'\right\rangle_{\Gamma_f}. \end{aligned} \tag{4.41}$$

Combining (4.35)–(4.41) gives (4.34). This completes the proof of the lemma. ∎

Remark 4.1. Applying Taylor expansion and omitting higher order terms gives

$$\begin{aligned} a_h^t(w, v) \approx & (\nabla w, \nabla v)_\Omega - \left\langle D_n w, v\right\rangle_{\Gamma_\Omega} - \left\langle D_n v, w\right\rangle_{\Gamma_\Omega} + \beta h^{-1}\left\langle w, v\right\rangle_{\Gamma_\Omega} \\ & - t\left(\left\langle D_n v, \nabla w\cdot\boldsymbol{\theta}\right\rangle_{\Gamma_\Omega} + \beta h^{-1}\left\langle \nabla w\cdot\boldsymbol{\theta}, v\right\rangle_{\Gamma_\Omega} + \beta h^{-1}\left\langle \nabla v\cdot\boldsymbol{\theta}, w\right\rangle_{\Gamma_\Omega}\right). \end{aligned} \tag{4.42}$$

Taking the derivative with respect to t in (4.42) and multiplying the result by -1 also gives (4.34).

Remark 4.2. We note that here the modified shape derivative $D_{\Omega,\boldsymbol{\theta}}\tilde{\mathcal{L}}_h(\Omega)$ is also exact for the discrete formulation. However, comparing to the discrete SD in (4.22) the boundary SD formula in (4.34) is more simple. Moreover, since the shape derivative only has surface forms on the free boundary, it enjoys the flexibility for the boundary type shape derivative.

5. Optimization algorithms. The objective now is to find the vector field $\boldsymbol{\theta} : \hat{\Omega} \to \hat{\Omega}$ such that the cost functional decreases the fastest along that direction. To this end we consider the following constrained minimization problem: find $\boldsymbol{\beta} \in H^1(\Omega)^d$ such that

$$\boldsymbol{\beta} = \underset{\substack{\|\boldsymbol{\theta}\|_{H^1(\hat{\Omega})}=1 \\ \boldsymbol{\theta}=0 \text{ on } \Gamma_f}}{\operatorname{argmin}} D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega, u, p). \tag{5.1}$$

Define the corresponding Lagrangian

$$\mathcal{K}(\boldsymbol{\theta}, \lambda) = D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega, u, p) + \lambda \left(\|\boldsymbol{\theta}\|^2_{H^1(\hat{\Omega})} - 1 \right).$$

From remark 4.1 in [16], an equivalent formulation of (5.1) renders to find $\tilde{\boldsymbol{\beta}} \in H^1_0(\hat{\Omega})^d$ such that

$$(\tilde{\boldsymbol{\beta}}, \boldsymbol{\theta})_{H^1(\hat{\Omega})} = -D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega, u, p) \quad \forall \boldsymbol{\theta} \in H^1_0(\hat{\Omega})^d, \tag{5.2}$$

where $\tilde{\boldsymbol{\beta}} = 2\lambda\boldsymbol{\beta}$ and $\lambda = \dfrac{\|\tilde{\boldsymbol{\beta}}\|_{H^1(\hat{\Omega})^d}}{2}$. Then it is easy to see that by taking $\boldsymbol{\theta} = \boldsymbol{\beta}$

$$D_{\Omega,\boldsymbol{\beta}}\mathcal{L} = -(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta})_{H^1(\hat{\Omega})^d} = -\|\tilde{\boldsymbol{\beta}}\|_{H^1_0(\hat{\Omega})^d} < 0, \tag{5.3}$$

which guarantees that $\boldsymbol{\beta}$ is a descent direction.

The following Hadamard Lemma indicates that under certain regularity the variational problem (5.2) is equivalent to an interface problem. See Theorem 2.27 and detailed definitions of function spaces in [36].

Lemma 5.1 (Hadamard). If $\mathcal{L}(\Omega)$ is shape differentiable at every element Ω of class $C^k, \Omega \subset \hat{\Omega}$. Furthermore, assume that $\partial\Omega$ is of class C^{k-1}. Then there exists a scalar function $\mathcal{G}(\Gamma_\Omega) \subset \mathcal{D}^{-k}(\Gamma_\Omega)$ such that

$$D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega) = \int_{\Gamma_\Omega} \mathcal{G}\boldsymbol{\theta} \cdot \boldsymbol{n}\, ds. \tag{5.4}$$

Combining (5.2) and Lemma 5.1 immediately gives

$$(\nabla\tilde{\boldsymbol{\beta}}, \nabla\boldsymbol{\theta})_\Omega + (\tilde{\boldsymbol{\beta}}, \boldsymbol{\theta})_\Omega = -\int_{\Gamma_\Omega} \mathcal{G}\boldsymbol{\theta} \cdot \boldsymbol{n}\, ds. \tag{5.5}$$

In strong form, equation (5.5) is equivalent to the following interface problem for $\tilde{\boldsymbol{\beta}} \in H^1(\Omega)^d$,

$$-\triangle\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\beta}} = 0 \quad \text{in } \hat{\Omega}, \tag{5.6}$$

$$[\![D_n\tilde{\boldsymbol{\beta}}]\!] = -\mathcal{G}\boldsymbol{n} \quad \text{on } \Gamma_\Omega, \tag{5.7}$$

$$[\![\tilde{\boldsymbol{\beta}}]\!] = 0 \quad \text{on } \Gamma_\Omega, \tag{5.8}$$

$$\tilde{\boldsymbol{\beta}} = 0 \quad \text{on } \partial\hat{\Omega}. \tag{5.9}$$

Given that Γ_Ω is smooth and $\mathcal{G} \in H^{1/2}(\Gamma_\Omega)$, we also have the following regularity estimate:

$$\|\tilde{\boldsymbol{\beta}}\|_{H^1(\hat{\Omega})} + \|\tilde{\boldsymbol{\beta}}\|_{H^2(\hat{\Omega}\setminus\Gamma_\Omega)} \lesssim \|\mathcal{G}\|_{H^{1/2}(\Gamma_\Omega)}, \tag{5.10}$$

(see [22]) and hence $\tilde{\boldsymbol{\beta}} \in H^1(\hat{\Omega})^d \cap H^2(\hat{\Omega} \setminus \Gamma_\Omega)^d$.

Here we illustrate the algorithm based on the cost functional for the continuous SD. In numerics, the continuous SD can be directly replaced by the discrete or boundary SD.

5.1. Approximation of the shape derivative $\tilde{\boldsymbol{\beta}}$ using CutFEM. In this subsection, we use the CutFEM of the interface type [28] to obtain a numerical approximation for $\tilde{\boldsymbol{\beta}}$ in (5.5). The same mesh used for solving (u_h, p_h) will also be used here. No fitting of the mesh to Γ_Ω is required.

We firstly define the related finite element spaces. Given a closed interface $\Gamma \subset \hat{\Omega}$, define $\Omega_\Gamma^- \subset \hat{\Omega}$ to be the domain enclosed by Γ and define $\Omega_\Gamma^+ = \hat{\Omega} \setminus \Omega_\Gamma^-$. Also define $\Omega_h^\pm = \cup\{K \in \mathcal{T}, K \cap \Omega_\Gamma^\pm \neq \emptyset\}$. Finally, define the finite element spaces $V_h^+(\Omega_h^+)$ and $V_h^-(\Omega_h^-)$ by

$$V_h^+(\Omega_h^+) = \{v^+ \in H^1(\Omega_h^+) : v^+|_K \in P^1(K) \quad \forall K \cap \Omega_\Gamma^+ \neq \emptyset\},$$

and

$$V_h^-(\Omega_h^-) = \{v^- \in H^1(\Omega_h^-) : v^-|_K \in P^1(K) \quad \forall K \cap \Omega_\Gamma^- \neq \emptyset\}.$$

Note that $V_h^+(\Omega_h^+)$ and $V_h^-(\Omega_h^-)$ are both defined on "cut" elements $K \in \mathcal{T}$ such that $K \cap \Gamma \neq \emptyset$. When there is no risk of ambiguity, we remove $(\Omega_h^\pm)$ in the finite element space notations.

The finite element solution for $\tilde{\boldsymbol{\beta}}$ is then set to find $\boldsymbol{\beta}_h := (\boldsymbol{\beta}_h^+, \boldsymbol{\beta}_h^-) \in V_h^+ \times V_h^-$ such that

$$b_0(\boldsymbol{\beta}_h, \boldsymbol{\theta}) + j(\boldsymbol{\beta}_h, \boldsymbol{\theta}) = l_1(\boldsymbol{\theta}) \quad \forall \boldsymbol{\theta} \in V_h^+ \times V_h^- \tag{5.11}$$

where

(5.12)
$$\begin{aligned} b_0(\boldsymbol{\beta}_h, \boldsymbol{\theta}) = (\nabla\boldsymbol{\beta}_h^+, \nabla\boldsymbol{\theta}^+)_{\Omega_\Gamma^+} + (\nabla\boldsymbol{\beta}_h^-, \nabla\boldsymbol{\theta}^-)_{\Omega_\Gamma^-} - \langle\{D_n\boldsymbol{\beta}_h\}, [\![\boldsymbol{\theta}]\!]\rangle_\Gamma - \langle D_n\boldsymbol{\beta}_h, \boldsymbol{\theta}\rangle_{\partial\hat{\Omega}} \\ - \langle\{D_n\boldsymbol{\theta}\}, [\![\boldsymbol{\beta}_h]\!]\rangle_\Gamma + \beta_1 h^{-1}\langle[\![\boldsymbol{\beta}_h]\!], [\![\boldsymbol{\theta}]\!]\rangle_\Gamma - \langle D_n\boldsymbol{\theta}, \boldsymbol{\beta}_h\rangle_{\partial\hat{\Omega}} + \beta_2 h^{-1}\langle\boldsymbol{\beta}_h, \boldsymbol{\theta}\rangle_{\partial\hat{\Omega}} \end{aligned}$$

(5.13)
$$j(\boldsymbol{\beta}_h, \boldsymbol{\theta}) = \gamma_1 h\left(\sum_{F\in\mathcal{E}_I(\Omega_h^+)}\int_F [\![D_n\boldsymbol{\beta}_h^+]\!][\![D_n\boldsymbol{\theta}^+]\!] + \sum_{F\in\mathcal{E}_I(\Omega_h^-)}\int_F [\![D_n\boldsymbol{\beta}_h^-]\!][\![D_n\boldsymbol{\theta}^-]\!]\right)$$

and

(5.14)
$$l_1(\boldsymbol{\theta}) = -D_{\Omega,\boldsymbol{\theta}}\mathcal{L}(\Omega, u_h, p_h) \quad \text{or} \quad -D_{\Omega,\boldsymbol{\theta}}\mathcal{L}_h(\Omega, u_h, p_h) \quad \text{or} \quad -D_{\Omega,\boldsymbol{\theta}}\tilde{\mathcal{L}}_h(\Omega, u_h, p_h),$$

where $\{D_n\boldsymbol{\theta}\}|_\Gamma := \dfrac{1}{2}\left(\nabla\boldsymbol{\theta}^+ + \nabla\boldsymbol{\theta}^-\right)\cdot\boldsymbol{n}_\Gamma$ is the arithmetic average operator where $\boldsymbol{n}_\Gamma$ is set to be the outer normal vector of Γ pointing from Ω_h^+ to Ω_h^-, and, $[\![\boldsymbol{\theta}]\!]|_\Gamma := \boldsymbol{\theta}^+ - \boldsymbol{\theta}^-$ is the jump operator.

5.2. Level set update. In this subsection, we update the free boundary Γ_Ω in the steepest descent direction (shape derivative) of $\boldsymbol{\beta}$. Our goal is to solve for the level set function $\phi(x+t\boldsymbol{\beta}(x),t)$ for the given $\boldsymbol{\beta}$ such that

$$\phi(x+t\boldsymbol{\beta}(x),t)=\phi(x,0) \quad \forall\, t \text{ and } \forall\, x\in\hat{\Omega}.$$

Taking the derivative with respect to t gives that

$$\nabla_x\phi\cdot\boldsymbol{\beta}+\frac{\partial\phi}{\partial t}=0 \quad \text{in}\,\hat{\Omega}. \tag{5.15}$$

This yields a Hamilton-Jacobi equation, if the nonlinear dependence of $\boldsymbol{\beta}$ on the optimization is accounted for. However for fixed vector field $\boldsymbol{\beta}$ this is simply an advection problem with a non-solenoidal transport field.

Remark 5.1. Note that we can simply choose the level set function at the initial stage as the distance function. However, after some evolution steps, the updated level set function no longer has this property. This can cause problems for accuracy of the numerical method if the magnitude of the gradient locally becomes very small or very large. Nevertheless, it is well known that the issue can be resolved by redefining ϕ regularly as the distance function while keeping the interface position fixed. In the numerical examples presented herein we did not notice any need for such re-distancing, since an advection stable scheme was used to propagate the interface.

To approximate (5.15), we use the Crank-Nicolson scheme in time combining with gradient penalty stabilization in space for the advection problem [19, 15]. We remain to use the same background mesh $\mathcal{T}$ for this step.

For the given Ω, let $\tau(\Omega,\boldsymbol{\beta}_h)=R*\dfrac{J(\Omega)}{\|\boldsymbol{\beta}_h\|_{H^1(\hat{\Omega})^d}}$, where $J(\Omega)$ is the cost functional defined in (2.6), R is the learning rate, and $\boldsymbol{\beta}_h$ is the solution to (5.11). We note that the steepest descent formula for τ is based on (5.3). Firstly, we divide $[0,\tau]$ into N equal length time steps and let $\delta t=\tau/N$ and $t_i=i\delta_t$ for $i=0,\cdots N$. Denote by $\phi_h^n=\phi_h(t_n)$. Given the initial level set ϕ_h^0, find $\phi_h^n\in V_h(\hat{\Omega})$ for $n=1,\cdots,N$ such that for all $w\in V_h(\hat{\Omega})$ there holds:

$$\left(\frac{\phi_h^n-\phi_h^{n-1}}{\delta t},w\right)_{\hat{\Omega}}+\frac{1}{\|\boldsymbol{\beta}_h\|_{H^1(\hat{\Omega})^d}}\left(\boldsymbol{\beta}_h\cdot\nabla\frac{\phi_h^n+\phi_h^{n-1}}{2},w\right)_{\hat{\Omega}}+r_h\left(\frac{\phi_h^n+\phi_h^{n-1}}{2},w\right)=0, \tag{5.16}$$

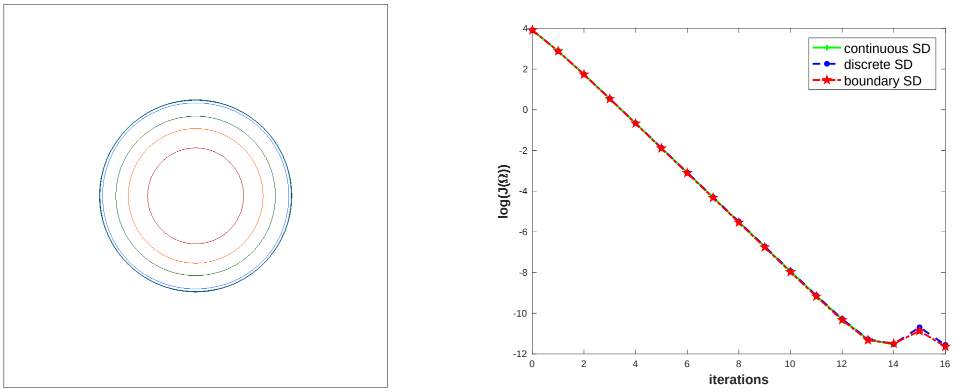
where

$$r_h(v,w)=\sum_{F\in\mathcal{E}_I(\hat{\Omega})}\gamma_2h^2\int_F[\![D_nv]\!][\![D_nw]\!]\,ds$$

with $\gamma_2>0$ is a positive parameter and $\mathcal{E}_I(\hat{\Omega})$ is the set of all interior facets in $\mathcal{T}$.

6. Numerical experiments. In the numerical experiments we mainly aim to compare the performances of the three different shape derivatives, i.e., continuous SD given in (4.14), the discrete SD given in (4.22), and the boundary SD given in (4.34).

A regular fixed background mesh of $\hat{\Omega}$ is used for all evolving PDE models. For all numerical experiments in this paper, we will use the unit square domain as the background domain, i.e., $\hat{\Omega}=[0,1]^2$. The background mesh is set as a uniform 100×100 crossed triangular mesh. The penalty parameters in (3.1) are chosen as $\gamma=0.1$ and $\beta=10$. And in (5.11), the parameters are chosen such that $\beta_1=\beta_2=10$ and $\gamma_1=1$. In (5.16), we chose $R=0.5$ or 1, $N=10$ and $\gamma_2=1$.



(a) level sets at iterations $0, 1, 2, 5$ and 10 (b) evolutions of cost functional

FIG. 2. *Example 6.1. Γ_{Ω^*} is a circle. Case 1. Initial level set as a circle.*

Example 6.1 (Circle). We recall the problem:

$$\begin{aligned} -\triangle u &= f \quad \text{in } \Omega^*, \\ u &= 0 \quad \text{on } \Gamma_{\Omega^*}, \\ u &= g_D,\ D_n u = g_N \quad \text{on } \Gamma_f. \end{aligned} \tag{6.1}$$

For this example, the free boundary Γ_{Ω^*} is the circle with radius $r_0 = 1/4$ and center being $(0.5, 0.5)$.

We choose to use the data (f, g_D, g_N) such that

$$f = -4/r, \quad g_D = 4r - 1 \text{ on } \partial\hat{\Omega}, \quad g_N = D_n u \text{ on } \partial\hat{\Omega}, \tag{6.2}$$

with $u = 4r - 1$ and $r = \sqrt{x^2 + y^2}$. We note that the choice for the boundary data is not unique and indeed there are infinitely many choices. Indeed, assuming $f \in L^2(\Omega^*)$ is given. For any $g_D \in H^{1/2}(\Gamma_f)$, there exits the so-called Dirichlet-Neumann mapping, $\mathcal{R} : g_D \in H^{1/2}(\Gamma_f) \to g_N \in H^{-1/2}(\Gamma_f)$ such that $g_N = D_n u$ and that u is the solution to

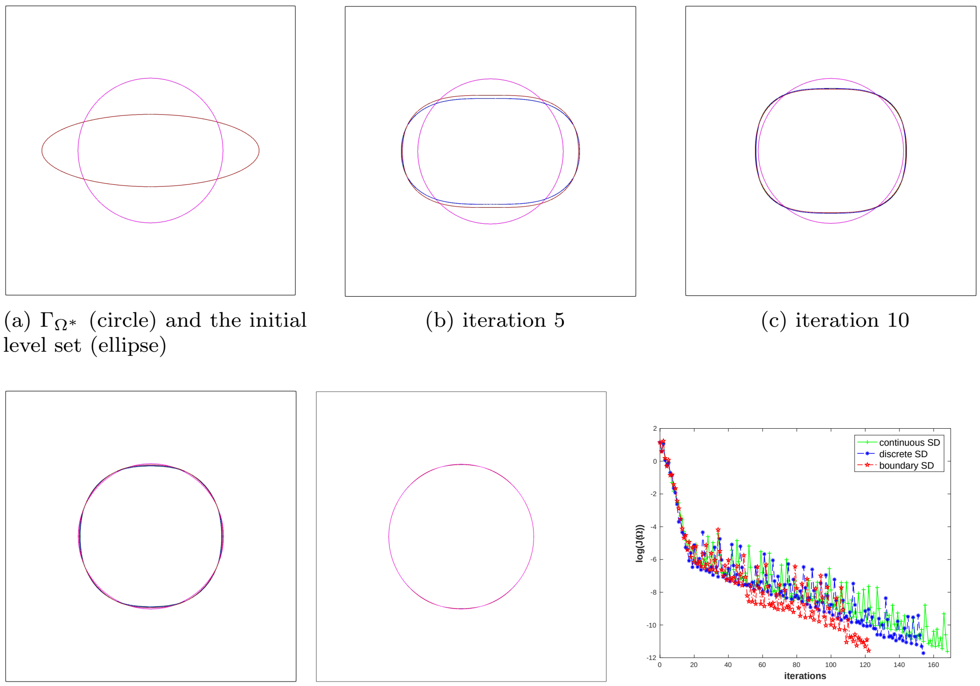
$$-\triangle u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \Gamma_{\Omega*}, \quad u = g_D \quad \text{on } \Gamma_f.$$

Therefore, for any g_D, we can use $(f, g_D, \mathcal{R}(g_D))$ as the given compatible data.

We start with a smaller circle (with same center $(0, 5, 0.5)$) as the initial free boundary (see the inner most circle in Figure 2a) that has the following level set function written in polar coordinates:

$$\phi(r, \theta) = -r + 1/8.$$

The stopping criteria is set such that $J(\Omega) \leq 1E - 5$. It takes 14, 16 and 16 iterations, respectively, using the continuous SD, discrete SD and boundary SD to reach the stopping criteria. In this case, the performances among all three shape derivatives are almost identical. Figure 2a shows the level sets at iterations $0, 1, 2, 5$ and 10 (from the inner most the to the outer most circles). The true level set is marked as magenta and is almost completely covered by the computed level set at step 10. The level set at iteration 0 is the initial given level set. At iteration 10, the



(a) Γ_{Ω^*} (circle) and the initial level set (ellipse)

(b) iteration 5

(c) iteration 10

(d) iteration 50

(e) final level sets

(f) evolutions of cost functional

FIG. 3. *Example 6.1: Γ_{Ω^*} is a circle. Case 2. Initial level set as an ellipse.*

computed level set almost coincides with the true level set function. Figure 2b shows the decreasing log rate of the cost functional $J(\Omega)$. In this case the cost functional converges at a fast and uniform rate for all three shape derivative.

We then test with an initial level set as an ellipse (see the red curve in Figure 3a):

$$\phi(x,y) = -\frac{(x-0.5)^2}{c_1^2} - \frac{(x-0.5)^2}{c_2^2} + 1, \text{ where } c_1 = 3/8, \text{ and } c_2 = 1/8.$$

With the same stopping criteria that $J(\Omega) \leq 1E-5$, it takes 169 , 155, and 123 iterations respectively for the continuous SD, discrete SD and boundary SD. Figure 3b–Figure 3d show the obtained level sets at iterations $5, 10$ and 50. The final converged computational level sets are given in Figure 3e. The level sets are marked with green for the continuous SD, blue for the discrete SD and red for the boundary SD. We again observe high coincidence among level sets computed by all SDs. Figure 3f compares the evolution of cost functionals. It is obvious to see two different convergence patterns for all cases: for about the first 20 iterations the cost functional is decreasing at a uniform fast rate with small oscillations and afterward is deceasing at a much slower rate with more severe oscillations.

If the initial level set is not properly chosen, the iterative procedure could require much more iterations to converge due to the very slow convergence in the second stage. Moreover, due to the nature of steepest descent method, iterations may stagnate at a local minimum.

We also note that the observed oscillations of the cost functional are natural since the pseudo time step is fixed. A more monotone behavior can be achieved if a line search is included. Furthermore, even though the discrete and boundary SDs are exact, the gradient $\boldsymbol{\beta}$ is not necessarily in the finite element space and, therefore, still requires approximation.

Example 6.2 (Ellipse). For this example, the free boundary Γ_{Ω^*} is an ellipse (see Figure 4a) with the following level set representation:

$$\phi(x,y) = -16(x-0.5)^2 - 64(y-0.5)^2 + 1.$$

We chose to use the data (f, g_D, g_N) such that $f \equiv 0$, $g_N = (\sin(x+y), \cos(x+y)) \cdot \boldsymbol{n}$ on Γ_f, and $g_D = \mathcal{R}^{-1}(g_N)$ where $\mathcal{R}^{-1}$ is the inverse mapping of the Dirichlet-Neumann mapping $\mathcal{R}$. Numerically, g_D is approximated by solving (3.4) on a 500×500 finer mesh.

We start with the following circle as the initial level set (see Figure 4a):

$$\phi(x,y) = -\sqrt{(x-0.6)^2 + (y-0.4)^2} + 1/6,$$

which has partial intersection with the true free boundary Γ_{Ω^*}. With the stopping criteria that $J(\Omega) \leq 1E-5$, it takes 120 , 154, and 146 iterations respectively for the continuous SD, discrete SD and boundary SD. Figure 4b–Figure 4d show the obtained level sets at iterations $5, 10$ and 50. The final computed level sets are given in Figure 4e. We again observe high coincidence among level sets computed by all SDs. Figure 4f compares the evolution of cost functional and similar phenomenons are observed to former examples. The number of iterations required to reach the stopping criteria also differs a significant amount due to its slow convergence rate and oscillating behavior in the second stage. In this case, unfortunately, the presenting algorithm is not able to yield significantly better level sets by simply running more iterations.

Example 6.3 (Lamé Square). For this example the free boundary Γ_{Ω^*} is a Lamé Square that has the following level set representation (see Figure 5a):

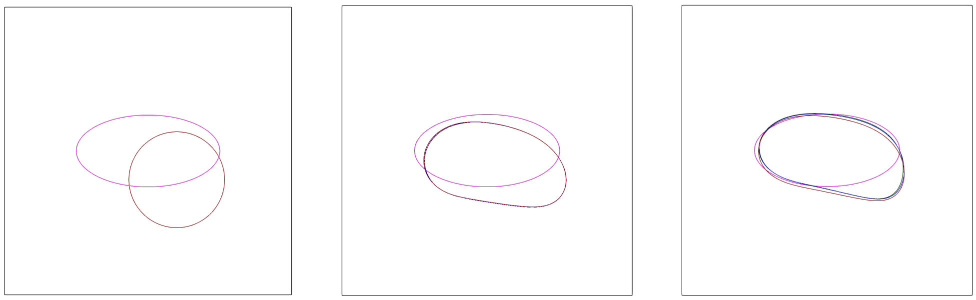
$$\phi(x,y) = -81(x-0.5)^n - 1296(y-0.5)^n + 1, \quad n = 4.$$

The level set becomes closer to a rectangle as the integer n increases. We chose the data (f, g_D, g_N) such that $f = 0$, $g_N = (5\sin(\theta), 5\cos(\theta)) \cdot \boldsymbol{n}$ where $\theta = \tan^{-1}((y - 0.5)/(x-0.5))$ and $g_D = \mathcal{R}^{-1}(g_N)$. Numerically, g_D is again approximated by solving (3.4) on a 500×500 finer mesh.

We start with the following circle as the initial level set (see Figure 5a)

$$\phi(x,y) = -\sqrt{(x-0.5)^2 + (y-0.5)^2} + 1/8.$$

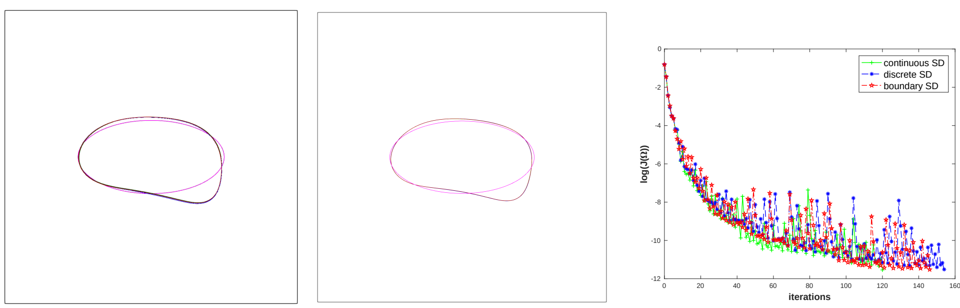
With the stopping criteria that $J(\Omega) \leq 5E-6$ with a maximal iteration number of 200, it takes 173, 174, and 200 iterations respectively using the continuous, discrete, and boundary SDs. Figure 5c – Figure 5d show the level sets at iterations $5, 10$ and 50. The final computed level sets are given in Figure 5e. In this case, the level sets produced by the continuous and discrete SDs are almost identical, however, are slightly different from those produced by the boundary SD. Figure 5f compares the evolution of cost functional. We observe different convergence patterns between the boundary SD and the rest. In the first 60 iterations, the cost functional based on the boundary SD decreases faster, however, for the remaining iterations its level sets remain steady.



(a) Γ_{Ω^*} (ellipse) and the initial level set (circle)

(b) iteration 5

(c) iteration 10



(d) iteration 50

(e) final level sets

(f) evolutions of cost functional

FIG. 4. *Example 6.2: Γ_{Ω^*} is an ellipse. Initial level set as a circle.*

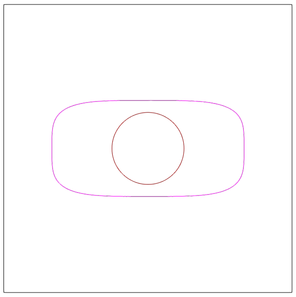
We also note that the final level sets in Figure 5e represent almost the best level sets we can achieve with the proposed algorithm. To illustrate, in Figure 6a we report the level set at the 1000th iteration for the discrete SD which barely shows any difference to its corresponding level set in Figure 5e. Figure 6b plots the evolution of the corresponding cost functional.

Example 6.4 (Topology change with merging). In this test, we aim to validate the capability of topology change for our algorithm. The free boundary Γ_{Ω^*} and the given data (f, g_D, g_N) are set to be the same as in Example 6.3. We choose the initial level set as two separate Lamé squares with the following level set functions (see Figure 7a):

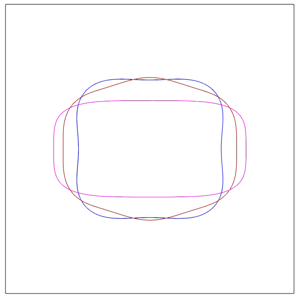
$$\phi(x, y) = \max\left(\phi_1(x, y), \phi_2(x, y)\right),$$

where $\phi_1(x, y) = 1-1296(x-0.32)^4-1296(y-0.5)^4$ and $\phi_2(x, y) = 1-1296(x-0.68)^4-1296(y-0.5)^4$. The stopping criteria is set the same that $J(\Omega) \le 5E-6$. It takes 271, 271, and 129 iterations for the respective continuous, discrete, and boundary SDs to reach the stopping criteria. Figure 7b -Figure 7e show the level sets at the respective iterations 10, 50 and 100 and the last iteration. We observe that the level set gradually merges into one simple connected shape for all SDs. The level sets obtained by all SDs are still almost identical. However, it takes significantly less iterations for the boundary SD as it converges slightly faster in the initial stage.

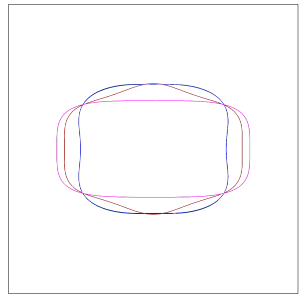
Example 6.5 (Doubly Connected Domain). In this example, the free boundary



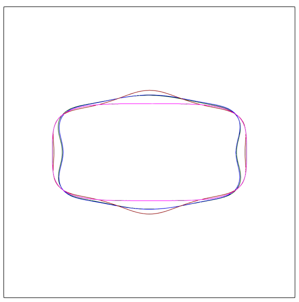
(a) Γ_{Ω^*} (Lamé Square) and the initial level set (circle)



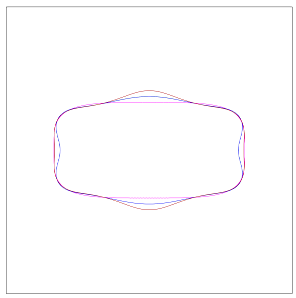
(b) iteration 5



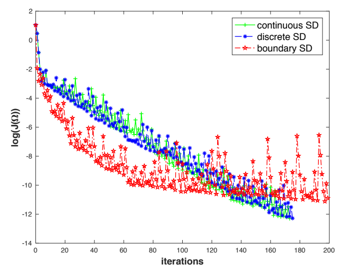
(c) iteration 10



(d) iteration 50

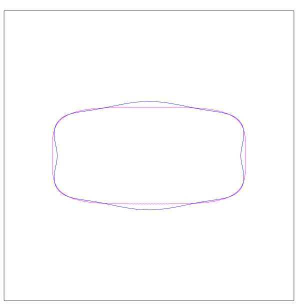


(e) final level sets

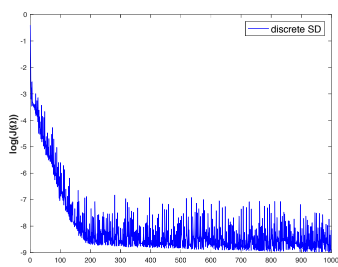


(f) evolutions of cost functional

FIG. 5. *Example 6.3: Γ_{Ω^*} as a Lamé Square. Initial level set as a circle.*



(a) iteration 1000



(b) evolutions of cost functional

FIG. 6. *Example 6.3: Γ_{Ω^*} as a Lamé Square. Initial level set as a circle.*

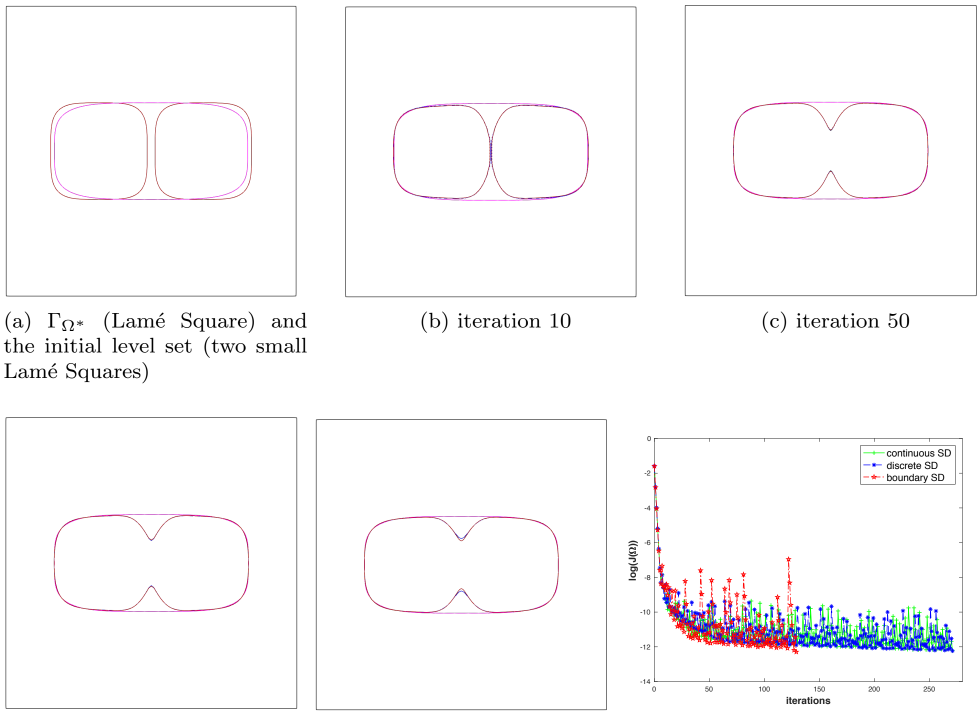
Γ_{Ω^*} is represented as two isolated circles (see Figure 8a):

$$\phi(x,y) = \max\left(0.15 - \sqrt{(x-0.2)^2 + (y-0.5)^2},\, 0.15 - \sqrt{(x-0.80)^2 + (y-0.5)^2}\right).$$

We start with the following simply connected Cassini oval as the initial level set (see Figure 8a)

$$\phi(x,y) = -(\hat{x}^2+\hat{y}^2)^2 + 2(\hat{x}^2-\hat{y}^2) - 1 + b^4, \quad \hat{x} = 3x - 1.5, \quad \hat{y} = 3y - 1.5, \quad b = 1.001.$$

The stopping criteria is set such that the maximal number of iterations not exceeds 300. We set the given data (f, g_D, g_N) such that $f = 0$, $g_N = (x-0.5, y-0.5)\cdot \boldsymbol{n}$



(a) Γ_{Ω^*} (Lamé Square) and the initial level set (two small Lamé Squares)

(b) iteration 10

(c) iteration 50

(d) iteration 100

(e) final level sets

(f) evolutions of cost functional

FIG. 7. *Example 6.4: Γ_{Ω^*} is a Lamé Square. Initial level set as two separated Lamé Squares.*

on Γ_f and $g_D = \mathcal{R}^{-1}$. Numerically, g_D is approximated again by solving (3.4) on a 500×500 finer mesh.

Figures Figure 8b–8e show the level sets at the respective iterations $50, 100, 200$ and 300. We observe that the Cassini oval gradually splits into two separate symmetric parts. Figure 8f compares the evolution of cost functional for the first 100 iterations. We observe that the convergence for this example is extremely slow which is likely due to the sharp angles (non-smoothness) evolved due to splitting. The results generated by the three SDs are again very similar.

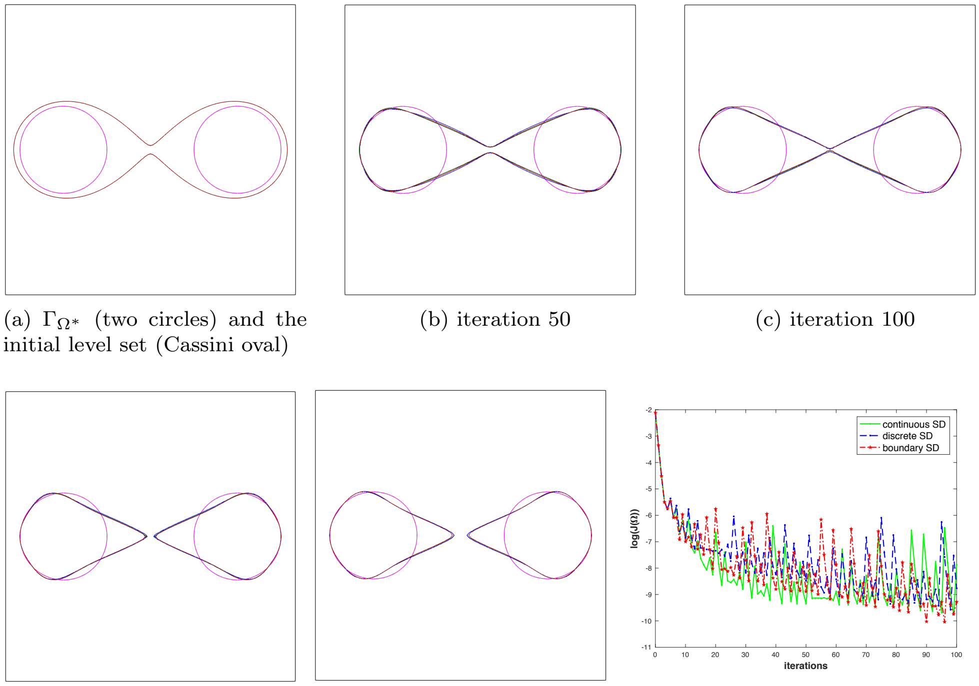
For all the numerical examples, we note that even the cost functionals exhibit oscillations in the second stage, the evolution of level sets remains relatively steady. We also observe that when the level sets involve non-smooth boundary, the convergence can be very slow.

7. Appendix.

Proof of Lemma 4.1.

Proof. Through a change of variable, we have

$$\int_{\Omega_t(\boldsymbol{\theta})} \phi(x,t)\,dx = \int_{\Omega} \phi \circ T_{t,\boldsymbol{\theta}} \mu_t\,dx = \int_{\Omega} \phi(x(t),t)\mu(t)\,dx$$



(a) Γ_{Ω^*} (two circles) and the initial level set (Cassini oval)

(b) iteration 50

(c) iteration 100

(d) iteration 200

(e) iteration 300

(f) evolutions of cost functional

FIG. 8. *Example 6.5: Γ_{Ω^*} as two separate circles. Initial level set as one simply connected Cassini oval.*

where $\mu(t) = \det(\nabla T_{t,\boldsymbol{\theta}})$ and $x(t) = x + t\boldsymbol{\theta}(x)$. Note that $\mu(0) = 1$. By definition,

$$
\begin{aligned}
D_{\Omega,\boldsymbol{\theta}} \int_\Omega \phi\, dx &= \lim_{t\to 0} \frac{1}{t}\left(\int_{\Omega_t(\boldsymbol{\theta})} \phi(x,t)\, dx - \int_\Omega \phi(x,0)\, dx \right) \\
&= \lim_{t\to 0} \int_\Omega \frac{1}{t}\left(\phi(x(t),t)\mu_t - \phi(x,0)\mu_0\right)\, dx \qquad (7.1)\\
&= \int_\Omega \dot{\phi}(x,0) dx + \int_\Omega \phi(x,0)\nabla\cdot\boldsymbol{\theta} dx
\end{aligned}
$$

where we have used the fact that (see Example 3.1 in [25])

$$\lim_{t\to 0} \frac{1}{t}(\mu(t) - \mu(0)) = \nabla\cdot\boldsymbol{\theta}.$$

To prove the second part of (4.7), we have that

$$\int_{\Gamma_{\Omega_t(\boldsymbol{\theta})}} \phi(x,t)\, dx = \int_{\Gamma_\Omega} \phi\circ T_{t,\boldsymbol{\theta}}\omega(t)\, dx = \int_{\Gamma_\Omega} \phi(x(t),t)\omega(t)\, dx$$

where $\omega(t) = \mu(t)|(\nabla T_{t,\boldsymbol{\theta}})^{-t}\cdot\boldsymbol{n}|$. Note that $\omega(0) = 1$. Finally, combining the fact that

$$\lim_{t\to 0} \frac{1}{t}(\omega(t) - \omega(0)) = \nabla\cdot\boldsymbol{\theta} - (\nabla\boldsymbol{\theta}\cdot\boldsymbol{n})\cdot\boldsymbol{n}$$

gives the second part of (4.7). This completes the proof of the lemma. ∎

Proof of Lemma 4.2.

Proof. By a change of variables, we have

$$
\begin{aligned}
&\lim_{t\to 0}\frac{1}{t}\left(\int_{\Omega_t(\boldsymbol{\theta})}\nabla w(x,t)\cdot\nabla v(x,t)\,dx-\int_{\Omega}\nabla w(x,0)\cdot\nabla v(x,0)\,dx\right)\\
&\quad=\lim_{t\to 0}\frac{1}{t}\left(\int_{\Omega}((\nabla w\circ T_t)\cdot(\nabla v\circ T_t)\mu(t)\,dx-\int_{\Omega}\nabla w(x,0)\cdot\nabla v(x,0)\,dx\right)\\
&\quad=\lim_{t\to 0}\frac{1}{t}\left(\int_{\Omega}(A(t)\cdot\nabla(w\circ T_t))\cdot\nabla(v\circ T_t)\,dx-\int_{\Omega}\nabla w\cdot\nabla v\,dx\right)\\
&\quad=\int_{\Omega}(A'(t)\cdot\nabla w)\cdot\nabla v+\nabla\dot{w}\cdot\nabla v+\nabla\dot{v}\cdot\nabla w\,dx,
\end{aligned}
\tag{7.2}
$$

where we used the chain rule

$$
(\nabla u)\circ T_t=\nabla T_t^{-t}\cdot\nabla(u\circ T_t)
\tag{7.3}
$$

and introduced $A(t)$ and its derivative

$$
A(t)=\mu(t)\nabla T_t^{-1}(\nabla T_t)^{-t},\qquad A'(t)=\nabla\cdot\boldsymbol{\theta}I-S(\boldsymbol{\theta}),
\tag{7.4}
$$

and finally we employed the product rule. This completes the proof of the lemma. ∎

Proof of Lemma 4.3.

Proof. Firstly by a change of variable we have

$$
\begin{aligned}
\int_{\Gamma_{\Omega_t}}\nabla w(x,t)\cdot\boldsymbol{n}_t v(x,t)\,ds&=\int_{\Gamma_\Omega}(\nabla w\circ T_t)\cdot(\boldsymbol{n}_t\circ T_t)(v\circ T_t)\omega(t)\,ds\\
&=\int_{\Gamma_\Omega}(\nabla T_t^{-t}\cdot\nabla(w\circ T_t))\cdot(\boldsymbol{n}_t\circ T_t)(v\circ T_t)\omega(t)\,ds.
\end{aligned}
\tag{7.5}
$$

From Theorem 4.4 in [25] it holds that

$$
\boldsymbol{n}_t\circ T_t=\frac{\nabla T_t^{-t}\cdot\boldsymbol{n}}{|\nabla T_t^{-t}\cdot\boldsymbol{n}|}.
$$

Recall that $\omega_t=\mu(t)|\nabla T_t^{-t}\cdot\boldsymbol{n}|$ and $A(t)=\mu(t)\nabla T_t^{-1}(\nabla T_t)^{-t}$. By a direct calculation together with (7.3) we have

$$
\int_{\Gamma_{\Omega_t}}(\nabla w(x,t)\cdot\boldsymbol{n}_t)v(x,t)\,ds=\int_{\Gamma_\Omega}(A(t)\cdot\nabla(w\circ T_t))\cdot\boldsymbol{n}(v\circ T_t)\,ds
\tag{7.6}
$$

Finally, combing (7.6) and (7.4) gives

$$
\begin{aligned}
&D_{\Omega,\boldsymbol{\theta}}\int_{\Gamma_\Omega}\nabla w\cdot\boldsymbol{n}v\,ds=\int_{\Gamma_\Omega}(A'(t)\cdot(\nabla w\cdot\boldsymbol{n})v\;+(\nabla\dot{w}\cdot\boldsymbol{n})v\,ds+(\nabla w\cdot\boldsymbol{n})\dot{v}\,ds\\
&=\int_{\Gamma_\Omega}((\nabla\cdot\boldsymbol{\theta})(\nabla w\cdot\boldsymbol{n})v-(S(\boldsymbol{\theta})\cdot\nabla w)\cdot\boldsymbol{n}v+(\nabla w\cdot\boldsymbol{n})\dot{v}\,ds+(\nabla\dot{w}\cdot\boldsymbol{n})v\,ds.
\end{aligned}
\tag{7.7}
$$

This completes the proof of the lemma. ∎

Proof of Lemma 4.4.

Proof. By the assumption that T_t is smooth, using similar arguments in Lemma 4.1 and Lemma 4.2 gives

$$
\begin{aligned}
&\int_{F^t} [\![\nabla w \cdot \boldsymbol{n}_t]\!][\![\nabla v \cdot \boldsymbol{n}_t]\!]\, ds \\
&\quad = \int_F [\![\nabla w \circ T^t \cdot (\boldsymbol{n}_t \circ T_t)]\!][\![\nabla v \circ T^t \cdot (\boldsymbol{n}_t \circ T_t)]\!] \omega(t)\, ds \\
&\quad = \int_F [\![A(t)\nabla(w \circ T^t) \cdot \boldsymbol{n}]\!][\![A(t)\nabla(v \circ T^t) \cdot \boldsymbol{n}]\!] \omega^{-1}(t)\, ds
\end{aligned} \tag{7.8}
$$

Applying the product rule, we then have that

$$
\begin{aligned}
&D_{\Omega,\boldsymbol{\theta}} \int_F [\![\nabla w \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!]\, ds \\
&= \int_F [\![A'(0)\nabla w \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!] + [\![A'(0)\nabla v \cdot \boldsymbol{n}]\!][\![\nabla w \cdot \boldsymbol{n}]\!] \\
&\quad + \int_F [\![\nabla w \cdot \boldsymbol{n}]\!][\![\nabla \dot{v} \cdot \boldsymbol{n}]\!] + [\![\nabla v \cdot \boldsymbol{n}]\!][\![\nabla \dot{w} \cdot \boldsymbol{n}]\!]\, ds \\
&\quad - \int_F [\![\nabla w \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!]\, \omega'(0) ds \\
&= \int_F [\![(\nabla \cdot \boldsymbol{\theta})\nabla w \cdot \boldsymbol{n} - S(\theta) \cdot \nabla w \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!]\, ds + \int_F [\![\nabla \dot{w} \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!]\, ds \\
&\quad + \int_F [\![(\nabla \cdot \boldsymbol{\theta})\nabla v \cdot \boldsymbol{n} - S(\theta) \cdot \nabla v \cdot \boldsymbol{n}]\!][\![\nabla w \cdot \boldsymbol{n}]\!]\, ds + \int_F [\![\nabla w \cdot \boldsymbol{n}]\!][\![\nabla \dot{v} \cdot \boldsymbol{n}]\!]\, ds \\
&\quad - \int_F [\![\nabla w \cdot \boldsymbol{n}]\!][\![\nabla v \cdot \boldsymbol{n}]\!]\, (\nabla \cdot \boldsymbol{\theta} - (\nabla \boldsymbol{\theta} \cdot \boldsymbol{n}) \cdot \boldsymbol{n}) ds.
\end{aligned} \tag{7.9}
$$

This completes the proof of Lemma 4.4. ∎

REFERENCES

[1] L. Afraites, M. Dambrine, K. Eppler, and D. Kateb, *Detecting perfectly insulated obstacles by shape optimization techniques of order two*, Discrete & Continuous Dynamical Systems-B, 8 (2007), pp. 389–416, https://doi.org/10.3934/dcdsb.2007.8.389.

[2] G. Allaire, F. Jouve, and A.-M. Toader, *A level-set method for shape optimization*, C. R. Math. Acad. Sci. Paris, 334 (2002), pp. 1125–1130, https://doi.org/10.1016/S1631-073X(02)02412-3.

[3] G. Allaire, F. Jouve, and A.-M. Toader, *Structural optimization using sensitivity analysis and a level-set method*, J. Comput. Phys., 194 (2004), pp. 363–393, https://doi.org/10.1016/j.jcp.2003.09.032.

[4] N. M. Atallah, C. Canuto, and G. Scovazzi, *Analysis of the shifted boundary method for the Poisson problem in general domains with corners*, Math. Comp., (2021), https://doi.org/10.1090/mcom/3641.

[5] A. Bernland, E. Wadbro, and M. Berggren, *Acoustic shape optimization using cut finite elements*, International Journal for Numerical Methods in Engineering, 113 (2018), pp. 432–449, https://doi.org/10.1002/nme.5621.

[6] F. Bouchon, S. Clain, and R. Touzani, *Numerical solution of the free boundary Bernoulli problem using a level set formulation*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 3934–3948, https://doi.org/10.1016/j.cma.2004.09.008.

[7] F. Bouchon, S. Clain, and R. Touzani, *A perturbation method for the numerical solution of the Bernoulli problem*, J. Comput. Math., 26 (2008), pp. 23–36, http://www.jstor.org/stable/43693422.

[8] L. Bourgeois and J. Dardé, *A quasi-reversibility approach to solve the inverse obstacle problem*, Inverse Probl. Imaging, 4 (2010), pp. 351–377, https://doi.org/10.3934/ipi.2010.4.351.
[9] L. Bourgeois and J. Dardé, *The "exterior approach" to solve the inverse obstacle problem for the Stokes system*, Inverse Probl. Imaging, 8 (2014), pp. 23–51, https://doi.org/10.3934/ipi.2014.8.23.
[10] J. H. Bramble, T. Dupont, and V. Thomée, *Projection methods for Dirichlet's problem in approximating polygonal domains with boundary-value corrections*, Math. Comp., 26 (1972), pp. 869–879, https://doi.org/10.2307/2005869.
[11] M. Burger, *A level set method for inverse problems*, Inverse Problems, 17 (2001), pp. 1327–1355, https://doi.org/10.1088/0266-5611/17/5/307.
[12] M. Burger, *Levenberg-Marquardt level set methods for inverse obstacle problems*, Inverse Problems, 20 (2004), pp. 259–282, https://doi.org/10.1088/0266-5611/20/1/016.
[13] M. Burger and S. J. Osher, *A survey on level set methods for inverse problems and optimal design*, European journal of applied mathematics, 16 (2005), pp. 263–301, https://doi.org/10.1017/S0956792505006182.
[14] E. Burman, *Ghost penalty*, Comptes Rendus Mathematique, 348 (2010), pp. 1217–1220, https://doi.org/10.1016/j.crma.2010.10.006.
[15] E. Burman, *Crank-Nicolson finite element methods using symmetric stabilization with an application to optimal control problems subject to transient advection-diffusion equations*, Commun. Math. Sci., 9 (2011), pp. 319–329, https://doi.org/10.4310/CMS.2011.v9.n1.a16.
[16] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson, *A cut finite element method for the Bernoulli free boundary value problem*, Comput. Methods Appl. Mech. Engrg., 317 (2017), pp. 598–618, https://doi.org/10.1016/j.cma.2016.12.021.
[17] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson, *Shape optimization using the cut finite element method*, Computer Methods in Applied Mechanics and Engineering, 328 (2018), pp. 242–261, https://doi.org/10.1016/j.cma.2017.09.005.
[18] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson, *Cut topology optimization for linear elasticity with coupling to parametric nondesign domain regions*, Comput. Methods Appl. Mech. Engrg., 350 (2019), pp. 462–479, https://doi.org/10.1016/j.cma.2019.03.016.
[19] E. Burman and M. A. Fernández, *Finite element methods with symmetric stabilization for the transient convection-diffusion-reaction equation*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 2508–2519, https://doi.org/10.1016/j.cma.2009.02.011.
[20] E. Burman, P. Hansbo, and M. G. Larson, *A cut finite element method with boundary value correction*, Math. Comp., 87 (2018), pp. 633–657, https://doi.org/10.1090/mcom/3240.
[21] E. Burman, P. Hansbo, and M. G. Larson, *Dirichlet boundary value correction using lagrange multipliers*, BIT Numerical Mathematics, 60 (2020), pp. 235–260, https://doi.org/10.1007/s10543-019-00773-4.
[22] Z. Chen and J. Zou, *Finite element methods and their convergence for elliptic and parabolic interface problems*, Numer. Math., 79 (1998), pp. 175–202, https://doi.org/10.1007/s002110050336.
[23] J. Cheung, M. Perego, P. Bochev, and M. Gunzburger, *Optimally accurate higher-order finite element methods for polytopial approximations of domains with smooth boundaries*, Mathematics of Computation, 88 (2019), pp. 2187–2219, https://doi.org/10.1090/mcom/3415.
[24] D. Colton and R. Kress, *Looking back on inverse scattering theory*, SIAM Review, 60 (2018), pp. 779–807, https://doi.org/10.1137/17M1144763.
[25] M. C. Delfour and J.-P. Zolésio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, SIAM, 2011, https://doi.org/10.1137/1.9780898719826.
[26] T. Dupont, J. Guzman, and R. Scott, *Obtaining higher-order Galerkin accuracy when the boundary is polygonally approximated*, arXiv e-prints, (2020), arXiv:2001.03082.
[27] J. Hadamard, *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, vol. 33, Imprimerie nationale, 1908.
[28] A. Hansbo and P. Hansbo, *An unfitted finite element method based on Nitsche's method for elliptic interface problems*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 5537–5552, https://doi.org/10.1016/S0045-7825(02)00524-8.
[29] R. Hiptmair and A. Paganini, *Shape optimization by pursuing diffeomorphisms*, Computational Methods in Applied Mathematics, 15 (2015), pp. 291–305, https://doi.org/10.1515/cmam-2015-0013.
[30] R. Hiptmair, A. Paganini, and S. Sargheini, *Comparison of approximate shape gradients*, BIT Numerical Mathematics, 55 (2015), pp. 459–485, https://doi.org/10.1007/s10543-014-0515-z.

[31] A. Laurain and K. Sturm, *Distributed shape derivative via averaged adjoint method and applications*, ESAIM: Mathematical Modelling and Numerical Analysis, 50 (2016), pp. 1241–1267, https://doi.org/10.1051/m2an/2015075.
[32] A. Main and G. Scovazzi, *The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems*, Journal of Computational Physics, 372 (2018), pp. 972–995, https://doi.org/10.1016/j.jcp.2017.10.026.
[33] J. Nitsche, *Über ein variationsprinzip zur lösung von Dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind*, in Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 36, Springer, 1971, pp. 9–15, https://doi.org/10.1007/BF02995904.
[34] S. Osher and R. P. Fedkiw, *Level set methods: an overview and some recent results*, Journal of Computational physics, 169 (2001), pp. 463–502, https://doi.org/10.1006/jcph.2000.6636.
[35] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A pde-based fast local level set method*, Journal of computational physics, 155 (1999), pp. 410–438, https://doi.org/10.1006/jcph.1999.6345.
[36] J. Sokoł owski and J.-P. Zolésio, *Introduction to shape optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992, https://doi.org/10.1007/978-3-642-58106-9.
[37] T. Strouboulis, I. Babuška, and K. Copps, *The design and analysis of the generalized finite element method*, Computer methods in applied mechanics and engineering, 181 (2000), pp. 43–69, https://doi.org/10.1016/S0045-7825(99)00072-9.
[38] C. H. Villanueva and K. Maute, *Cutfem topology optimization of 3d laminar incompressible flow problems*, Computer Methods in Applied Mechanics and Engineering, 320 (2017), pp. 444–473, https://doi.org/10.1016/j.cma.2017.03.007.
[39] M. Y. Wang, X. Wang, and D. Guo, *A level set method for structural topology optimization*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246, https://doi.org/10.1016/S0045-7825(02)00559-5.
[40] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, *Immersed finite element method*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 2051–2067, https://doi.org/10.1016/j.cma.2003.12.044.