Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

Numerical solution to the exterior Bernoulli problem using the Dirichlet-Robin energy gap cost functional approach in two and three dimensions

Julius Fergy Tiongson Rabago[†]

Faculty of Mathematics and Physics, Kanazawa University, Kakumamachi, Kanazawa, 920-1192, Ishikawa, Japan.

[†]E-mail: jfrabago@gmail.com

Abstract

The exterior Bernoulli problem – a prototype stationary free boundary problem – is rephrased into a shape optimization setting using an energy-gap type cost functional that is subject to two auxiliary problems: a pure Dirichlet problem and a mixed Dirichlet-Robin boundary value problem. It is demonstrated here that depending on what method is used, the shape gradient of the cost functional may appear in a different form. The dissimilarity in structure comes from the way the adjoint variable was utilized in the computation – then resulting to a different adjoint problem. The shape derivative is first obtained via Delfour-Zolésio's minimax formulation, and then by using the weak form of the Eulerian derivative of the states coupled with the adjoint method. The latter approach is accomplished by first showing the existence of the derivatives. A fast iterative scheme based on finite element method is then formulated to numerically solve the proposed shape optimization formulation. The feasibility of the method – highlighting its efficiency and practicality – is illustrated through numerical examples in two and three dimensions.

Keywords: Bernoulli problem, domain perturbation, free boundary, Lagrangian method, minimax formulation, shape optimization, shape derivative

MSC Classification: 49Q10 , 49K20 , 65K10

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

1 Introduction

Let A be a given open (non-empty) bounded and connected domain in $\mathbb{R}^d$, $d \in \{2, 3\}$, with a fixed boundary $\Gamma := \partial A$, and $\lambda < 0$ be a given fixed constant. The exterior Bernoulli free boundary problem is described as a problem of finding an open bounded connected domain $B \subset \mathbb{R}^d$, $B \neq \emptyset$, with a boundary $\Sigma := \partial B$, containing the closure of A, and an associated state function $u := u(\Omega)$, where $\Omega = B \setminus \overline{A}$, such that the following overdetermined boundary problem is satisfied:

$$-\Delta u = 0 \text{ in } \Omega, \qquad u = 1 \text{ on } \Gamma, \qquad u = 0 \quad \text{and} \quad \partial_{\mathbf{n}} u = \lambda \text{ on } \Sigma, \tag{1}$$

where $\partial_{\mathbf{n}} u := \nabla u \cdot \mathbf{n}$ denotes the (outward) normal derivative of u. For any $\lambda < 0$, problem (1) is known to admit a classical solution for simply connected bounded domain Ω. In addition, the shape solution Ω^* is unique for bounded convex domains A [1], and the free boundary Σ_* is $C^{2,\alpha}$ ($\alpha \in (0, 1)$) regular (see [2, Thm 1.1]). For more details about the qualitative properties of solutions to (1) and their numerical approximations, we refer the readers to [1] (see also [3] for a related Bernoulli problem with a geometric constraint).

The exterior Bernoulli problem (1) (a.k.a. the Alt-Caffarelli problem, see [4]) originates from the description of free surfaces for ideal fluids [5]. It also arises in numerous other applications, for example in the context of optimal design, electro chemistry, and electro statics [1, 6].

A well-established approach to solve a free boundary problem is to formulate it into a shape optimization setting. Such approach can be done in different ways. For example, one can choose one of the boundary conditions on the free boundary to obtain a well-posed state equation and then track the remaining boundary data in a least-squares sense (see [7–13]). Alternatively, one can consider an energy-gap type cost function which consists of two auxiliary states: one that is a solution of a pure Dirichlet problem and one that satisfies a mixed Dirichlet-Neumann problem (see [14–18]). Modifications of these classical formulations were recently offered in [19, 20], including a new boundary integral type cost functional minimization approach put forward in [21]. In these formulations, the mixed Dirichlet-Neumann problem is replaced by a mixed Robin-Neumann problem, and it was found through multifarious numerical experiments that such a modification provide some computational advantages over the classical formulations in solving the free boundary problem (1). One may also apply the so-called coupled complex boundary method [22] to write the overdetermined problem as an optimization problem with a complex valued state problem, see [23].

Motivated by the above-mentioned works, we consider here the following shape optimization reformulation of (1)

$$J(\Omega) = \frac{1}{2} \int_{\Omega} |\nabla(u_D - u_R)|^2 \, dx \quad \longrightarrow \quad \inf, \tag{2}$$

where the state variables u_D and u_R satisfy the following PDE systems

$$-\Delta u_D = 0 \text{ in } \Omega, \qquad u_D = 1 \text{ on } \Gamma, \qquad u_D = 0 \text{ on } \Sigma; \tag{3}$$

$$-\Delta u_R = 0 \text{ in } \Omega, \qquad u_R = 1 \text{ on } \Gamma, \qquad \partial_{\mathbf{n}} u_R + \beta u_R = \lambda \text{ on } \Sigma, \tag{4}$$

respectively, and $\beta := \beta(x) \geqslant 0$ is a fixed Lipschitz (non-identically zero) function in $\mathbb{R}^d$. For $\beta \equiv 0$ in (4), we recover the classical Kohn-Vogelius[1] method to (1) which has already been extensively studied in [14, 15, 18, 25]. By this connection, the minimization problem (2) may be viewed as a small generalization of the Kohn-Vogelius method.

Observe that the minimizer of J is the solution of (1). To validate this claim, let us consider the functional space

$$V(\Omega) = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma\}$$

endowed with the norm

$$\|\cdot\|_{V(\Omega)} := \left(\int_\Omega |\cdot|^2 \, dx + \int_\Sigma |\cdot|^2 \, ds \right)^{1/2}.$$

On the one hand, if $(\Omega, u(\Omega))$ satisfies (1), then $u_D = u_R = u$, and $u_R = 0$ and $\partial_{\mathbf{n}} u_R = \lambda$ on Σ; therefore, $J(\Omega) = 0$. On the other hand, since $u_D - u_R \in V(\Omega)$, the shape functional $J(\Omega)$ is well-defined. If $J(\Omega) = 0$, then $0 = \int_\Omega |\nabla(u_D - u_R)|^2 \, dx = -\int_\Sigma (\beta |u_R|^2 + u_R(\partial_{\mathbf{n}} u_D - \lambda)) \, ds$. This equation obviously holds when $u_R = 0$ on Σ, and so $\partial_{\mathbf{n}} u_R = \lambda$ on Σ. It is also true when $\partial_{\mathbf{n}} u_D = \lambda$ on Σ, from which it is evident that $u_R = 0$ on Σ. In fact, since the semi-norm $|\cdot|_{H^1(\Omega)}$ is a norm on $V(\Omega)$, then $J(\Omega) = 0$ indeed implies that $u_D = u_R$ and $u = u_D = u_R$ is a solution of (1). Hence, $(\Omega, u(\Omega))$ is a solution of (1) if and only if $J(\Omega) = 0$. Consequently, we see that solving (1) is equivalent to finding $(\Omega, u(\Omega))$ such that

$$J(\Omega) = \min_{\widetilde{\Omega}} J(\widetilde{\Omega}).^{2} \tag{5}$$

Because of the equivalence of the shape optimization problem (5) to the overdetermined boundary value problem (1), the existence of solution to the aforementioned shape problem is guaranteed. Nevertheless, the question of existence of optimal solution to (5) can be addressed rigorously using, for example, the ideas developed in [10, 26, 27] and apply the tools furnished, for instance, in [28–30].

Note that since (16) and (18) are uniquely solvable in $H^1(\Omega)$, one can define the map $\Omega \mapsto (u_D(\Omega), u_R(\Omega))$ whose graph is given by

$$\mathcal{F} = \{(\Omega, u_D, u_R) : \Omega \in \mathcal{O}_{\text{ad}},\ u_D(\Omega) \text{ and } u_R(\Omega) \text{ solves (16) and (18) on } \Omega\}.$$

[1] This case is often attributed to Kohn and Vogelius [24] since they were among the first to use the functional in the context of inverse problems.

[2] Here, it is understood that the minimization is carried out over some set of admissible domains. This admissible set will be specified at the beginning of Section 2.

Problem (5) is then equivalent to minimizing $J(\Omega) = J(\Omega, u_D, u_R)$ on $\mathcal{F}$. To prove the existence of solution to this minimization problem, the set $\mathcal{F}$ needs to be endowed with a topology for which it is compact and then prove that J is lower semi-continuous. We will not give the details of the proof here, but a similar approach used in [20] can be applied, at least for the case of two dimensions. For the three dimensional case, a more general approach using the concept of convergence of sets in the sense of Hausdorff [31] combined with (uniform) cone property (see [32]) can be used.

Further motivation of this work will be given in this paragraph. The shape optimization problem (5) was first studied in [33]. However, it lacks of any numerical investigation that could somehow evince the quality of the formulation when employed in practice. This fact clearly warrants further research on the present formulation. The main purpose of this investigation, therefore, is to give a numerical realization of (5). In the latter section of the paper, we will show that the present formulation actually provides a fast numerical approximation of the solution to (1) (this highlights the numerical part of this study) when solved via a Lagrangian-type first-order gradient-based iterative scheme. Accordingly, the structure of the so-called shape gradient is a requisite in this exposition. This expression can be obtained in various ways, which depends – among other requirements – on the degree of smoothness of the domain. For example, when the boundary of the domain is $C^{2,1}$ regular, one may apply the minimax formulation [34] (see also [35, Chap. 10.6]) to obtain the boundary integral representation of the shape derivative of J. The expression can also be obtained through the chain rule approach as done in [33]. Contrary to the minimax formulation, the said technique demands the knowledge of the shape derivative of the states which are quite tedious to establish, especially in the case of more complicated state constraints (see, e.g., [17] and [21, Appx. A]). The method developed in [34], however, naturally requires the use of appropriate adjoint state systems in order to characterize the shape derivative of the cost, thereby bypassing the use of the shape derivative of the states. It is worth to mention here that both the aforementioned approaches require a $C^{2,1}$ regularity of the domain (and of the deformation field) in order to derive the boundary integral expression of J. Nevertheless, it is possible to obtain the shape derivative of the cost with Ω belonging only to the $C^{1,1}$ class of domains. This can be done either through the method developed in [36] or by using the Eulerian derivative of the states[3] – the latter which we will demonstrate here with the help of the adjoint method.

To sum up, the main points and contributions of this note are as follows:

- to provide a rigorous derivation of the shape derivative of J along the lines of argumentation (by the minimax formulation) used in [34, Sec. 6] – bypassing the need to use the strong form of the shape derivative of the states;

[3] For a related study regarding the existence of such a derivative, but for the case of stationary nonlinear heat equation, one may refer to [37].

- to characterize the boundary integral form of the shape derivative of J – under the very mild $C^{1,1}$ regularity assumption on Ω – by means of only the weak form of the Eulerian derivative of the states coupled with the adjoint method;
- to carry out – in a fast and efficient manner – the numerical realization of (5) via a Lagrangian-type gradient-based numerical scheme using finite element methods (FEMs) without remeshing.

Additionally, in this investigation we want to demonstrate that the shape gradient of J may take a different structure depending on what approach is used to derive the expression (i.e., e.g., via the minimax formulation or through the Eulerian derivative of the state). The difference in structure essentially comes from the way the adjoint variable is utilized in the computation.

Notice in (4) that we have the freedom to choose the Robin parameter β, as long as it is non-negative and at least Lipschitz continuous. In our numerical experiments, we will consider taking β as the mean curvature κ of the free boundary Σ. Although it is not yet clear at this point the motivation in considering such choice, the matter will be made more transparent after the expression for the shape gradient of J is obtained; see Proposition 2.1.1 and Proposition 2.2.5 (refer also to Remark 6 for a related note). We also want to emphasize in advance that the said choice of β seems to provide fast approximation of the optimal shape solution to the shape problem (5) as observed in subsection 3.2. The main point, on the other hand, of the second objective stated above is to demonstrate how to apply the adjoint method to deal with the derivative of a Robin problem that appears naturally in the structure of the shape gradient that was obtained through the Eulerian derivative of the states. Meanwhile, a numerical realization of the present shape optimization problem (carried out here via FEM) is demanded to check the correctness of the shape derivatives and to showcase the feasibility and practicality of (2) in resolving (1). We mention in advance that, as opposed to previous works, we do not employ in our numerical scheme any remeshing process such as adaptive mesh refinement. We do this to evaluate to some extent the stability of our method.

The remaining part of the paper is outlined as follows. In the next section, we derive the first-order shape derivative of J by minimax formulation (subsections 2.1), and then via the Eulerian derivative of the states coupled with the adjoint method (subsection 2.2). In Section 3, we carry out the numerical realization of problem (5). First, we detail in subsection 3.1, the algorithm used to numerically solve (5), and then present some numerical tests in subsections 3.2–3.4. Since most of the previous studies only dealt with problems in two dimensions (except in [7, 23], and also in [38] where a Newton scheme with a Dirichlet energy functional was employed to solve a slightly more general Bernoulli free boundary problem), we also put our attention on testing the proposed method to three dimensional cases (see subsection 3.4). Finally, we give a short conclusion in Section 4.

2 Shape derivatives of the cost

Let us consider a hold-all set $U \subset \mathbb{R}^d$ (which is at least Lipschitz regular) strictly containing $\overline{\Omega}$ (the closure of Ω) and define T_t as the perturbation of the identity id given by the map

$$T_t = T_t(\boldsymbol{V}) = id + t\boldsymbol{V},$$

where $\boldsymbol{V}$ is a t-independent deformation field belonging to the admissible space

$$\boldsymbol{\Theta}^k := \{\boldsymbol{V} \in C^{k,1}(\overline{\Omega})^d \mid \boldsymbol{V} = \boldsymbol{0} \text{ on } \Gamma \cup \partial U\},^{4} \tag{6}$$

and k is a natural number (later on specified depending on what is needed). Here, and throughout the paper, t is assumed sufficiently small such that T_t is a diffeomorphism from $\Omega \in C^{k,1}$ onto its image. That is, the reference domain Ω and its perturbation Ω_t have the same topological structure and regularity under the transformation T_t. To be more specific, we let $\varepsilon > 0$ be a sufficiently small real number such that $[t \mapsto T_t] \in C^1(\mathcal{I}, C^{1,1}(\overline{U})^d)$, where $\mathcal{I} := [0, \varepsilon]$.

By definition, we also have the perturbation $\Sigma_t := T_t(\Sigma)$ and $\Gamma_t := T_t(\Gamma) \equiv \Gamma$. The latter identity is due to the fact that $\boldsymbol{V} = \boldsymbol{0}$ on Γ which means that, essentially, Γ remains a part of Ω_t for all t. Moreover, $\Omega_0 = \Omega$ and $\Sigma_0 = \Sigma$. Accordingly, the set of all admissible domains $\mathcal{O}_{ad}$ is given as follows

$$\mathcal{O}_{ad} = \left\{T_t(\boldsymbol{V})(\overline{\Omega}) \subset U \mid \Omega \in C^{k,1}, k \in \mathbb{N}, t \in \mathcal{I}, \boldsymbol{V} \in \boldsymbol{\Theta}^k\right\}. \tag{7}$$

The functional $J : \mathcal{O}_{ad} \to \mathbb{R}$ has a directional first-order Eulerian derivative at Ω in the direction of the field $\boldsymbol{V}$ if the limit

$$\lim_{t \searrow 0} \frac{J(\Omega_t) - J(\Omega)}{t} =: dJ(\Omega)[\boldsymbol{V}] \tag{8}$$

exists (cf. [35, Sec. 4.3.2, Eq. (3.6), p. 172] and [40, Def. 2.1]). If the map $\boldsymbol{V} \mapsto dJ(\Omega)[\boldsymbol{V}]$ is linear and continuous, then the shape functional J is shape differentiable at Ω. In this case, the map is referred to as the shape gradient of J at Ω.

Throughout the paper, for simplicity, we assume that β is a positive constant (unless otherwise stated). Moreover, we write $\boldsymbol{V} \cdot \mathbf{n}$ as V_n.

2.1 Delfour-Zolésio's minimax formulation approach

We derive here the shape derivative of J via a minimax formulation with $k = 2$ in (7).[5] Because we are considering domains of class of $C^{2,1}$, we can also characterized the shape derivative of J in terms of a boundary integral.

[4] For discussions about open sets of class $C^{k,l}$, $k \in \mathbb{N}$, $l \in (0, 1]$, see [39].

[5] A Lipschitz regularity for Γ is enough to establish the shape derivatives, however, for simplicity, we assume that Γ and Σ are both $C^{k,1}$ regular, where $k = 1$ or 2.

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

Proposition 2.1.1 Let $\Omega \in C^{2,1}$, $\boldsymbol{V} \in \boldsymbol{\Theta}^2$, u_D be the solution to (3), and u_R satisfy (4). Then, J is shape differentiable, and its shape derivative is $dJ(\Omega)[\boldsymbol{V}] = \int_\Sigma \mathcal{G}_\beta \mathbf{n} \cdot \boldsymbol{V}\, ds$, where the shape gradient $\mathcal{G}_\beta$ is given by[6]

$$\mathcal{G}_\beta := -\frac{1}{2}(\partial_{\mathbf{n}} u_D)^2 + \partial_\tau u_R \partial_\tau p_R - \partial_{\mathbf{n}} u_R (\kappa - \beta) p_R + \frac{1}{2}\left((\partial_{\mathbf{n}} u_R)^2 + (\partial_\tau u_R)^2\right), \tag{9}$$

and the adjoint state $p_R \in H^1(\Omega)$ satisfies the PDE system

$$-\Delta p_R = 0 \text{ in } \Omega, \qquad p_R = 0 \text{ on } \Gamma, \qquad \partial_{\mathbf{n}} p_R + \beta p_R = \beta u_R - \lambda \text{ on } \Sigma. \tag{10}$$

Moreover, at the shape solution Ω^* of (1), we have $\mathcal{G}_\beta \equiv 0$ on Σ_* and it holds that

$$dJ(\Omega^*)[\boldsymbol{V}] = 0, \quad \text{for all } \boldsymbol{V} \in \boldsymbol{\Theta}^2.$$

Before we prove the proposition, we first introduce a few additional notations. We denote by DT_t the Jacobian matrix of T_t and write the inverse and inverse transpose of this matrix by $(DT_t)^{-1}$ and $(DT_t)^{-\top}$, respectively. For convenience, we define

$$I_t := \det DT_t, \quad A_t := I_t(DT_t^{-1})(DT_t)^{-\top}, \quad \text{and} \quad B_t := I_t|(DT_t)^{-\top}\mathbf{n}|.$$

For $t \in \mathcal{I}$, I_t is positive. At $t = 0$, it is evident that $I_0 = 1$, $A_0 = id$, $B_0 = 1$. Moreover, on $\mathcal{I}$, the maps $t \mapsto I_t$, $t \mapsto A_t$, and $t \mapsto B_t$ are continuously differentiable. That is, for $t \in \mathcal{I}$, we have (see, e.g., [11, 36])

$$\begin{cases} [t \mapsto I_t] \in C^1(\mathcal{I}, C(\overline{\Omega})), \\ [t \mapsto A_t] \in C^1(\mathcal{I}, C(\overline{\Omega})^{d\times d}), \\ [t \mapsto B_t] \in C^1(\mathcal{I}, C(\Sigma)). \end{cases} \tag{11}$$

Additionally, $[t \mapsto I_t] \in C^1(\mathcal{I}, C^{0,1}(\overline{U}))$ and $[t \mapsto A_t] \in C(\mathcal{I}, C(\overline{U})^{d\times d})$. The derivatives of the above maps are respectively given as follows:

$$\begin{aligned} \frac{d}{dt} I_t\Big|_{t=0} &= \lim_{t\to 0} \frac{I_t - 1}{t} = \operatorname{div} \boldsymbol{V}, \\ \frac{d}{dt} A_t\Big|_{t=0} &= \lim_{t\to 0} \frac{A_t - id}{t} = (\operatorname{div} \boldsymbol{V}) id - D\boldsymbol{V} - (D\boldsymbol{V})^\top =: A, \\ \frac{d}{dt} B_t\Big|_{t=0} &= \lim_{t\to 0} \frac{B_t - 1}{t} = \operatorname{div}_\Sigma \boldsymbol{V} = \operatorname{div} \boldsymbol{V}\Big|_\Sigma - (D\boldsymbol{V}\mathbf{n}) \cdot \mathbf{n}, \end{aligned} \tag{12}$$

where $\operatorname{div}_\Sigma \boldsymbol{V}$ denotes the tangential divergence of the vector $\boldsymbol{V}$ on Σ.

For later use, we assume in addition that for $t \in \mathcal{I}$, we have

$$0 < \Lambda_1 \leqslant I_t \leqslant \Lambda_2 \qquad \text{and} \qquad 0 < \Lambda_3 |\xi|^2 \leqslant A_t \xi \cdot \xi \leqslant \Lambda_3 |\xi|^2, \tag{13}$$

[6] The vector τ here represents the unit tangent vector on Σ.

for all $\xi \in \mathbb{R}^d$, for some constants Λ_1, Λ_2, Λ_3, and Λ_4 ($\Lambda_1 < \Lambda_2$, $\Lambda_3 < \Lambda_4$). Moreover, for $u, v \in H^1(\Omega)$ and $t \in \mathcal{I}$, we introduce the following forms

$$a^t(u,v) = \int_\Omega A_t \nabla u \cdot \nabla v \, dx + \int_\Sigma B_t \beta u v \, ds, \quad \text{and} \quad b^t(v) = \int_\Sigma \lambda B_t v \, ds$$

For $t = 0$, we write $a_0(\cdot,\cdot) = a(\cdot,\cdot)$ and $b_0(\cdot) = b(\cdot)$, where

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx + \int_\Sigma \beta u v \, ds \quad \text{and} \quad b(v) = \int_\Sigma \lambda v \, ds,$$

respectively. Also, for brevity, we occasionally write

$$(\nabla u, \nabla v)_\Omega = \int_\Omega \nabla u \cdot \nabla v \, dx \quad \text{and} \quad \langle u, v \rangle_\Sigma = \int_\Sigma u v \, ds.$$

So, we may write $a(u,v) = (\nabla u, \nabla v)_\Omega + \langle \beta u, v \rangle_\Sigma$.

Lastly, as we do not care about exact estimates, we use in this paper the notation "$\lesssim$" which means that if $P \lesssim Q$, then we can find some constant $c > 0$ such that $P \leqslant cQ$. Of course, $Q \gtrsim P$ is defined as $P \lesssim Q$.

Proof of Proposition 2.1.1 Let us assume, for the mean time, that $\Omega \in C^{1,1}$ and $\boldsymbol{V} \in \boldsymbol{\Theta}^1$. To start, we observe that

$$J(\Omega) = J_D(\Omega) + J_{\mathrm{DR}}(\Omega) + J_R(\Omega),$$

where $J_D(\Omega) = \frac{1}{2}(\nabla u_D, \nabla u_D)_\Omega$, $J_{\mathrm{DR}}(\Omega) = -(\nabla u_D, \nabla u_R)_\Omega$, and $J_R(\Omega) = \frac{1}{2}(\nabla u_R, \nabla u_R)_\Omega$. By Green's formula, together with (3), we can actually write J_{DR} as $J_{\mathrm{DR}}(\Omega) = -\int_\Gamma \partial_{\mathbf{n}} u_R \, ds$. Since $\boldsymbol{V} = \boldsymbol{0}$ on Γ, the shape derivative of J_{DR} vanishes, and therefore does not contribute any value to the shape gradient. From this observation, we may disregard the functional J_{DR} and consider, instead of (5), the minimization problem $J(\Omega) = \min_{\widetilde{\Omega}} J_D(\widetilde{\Omega}) + \min_{\widetilde{\Omega}} J_R(\widetilde{\Omega})$. This tells us that, essentially, we have $dJ(\Omega)[\boldsymbol{V}] = dJ_D(\Omega)[\boldsymbol{V}] + dJ_R(\Omega)[\boldsymbol{V}]$. However, we keep J_{DR} in our argumentation for some technical reasons (see, e.g., Remark 1).

The proof which proceeds in four steps is now in order.

Step 1. We first construct appropriate functionals corresponding to J_D and J_R over the perturbed domain Ω_t. Taking into account the extra constraints $u_{Dt} = 1$ on Γ and $u_{Dt} = 0$ on Σ_t for $J_D(\Omega_t)$, and similarly, $u_{Rt} = 1$ on Γ for $J_R(\Omega_t)$, we introduce the functionals

$$G_D(t, \varphi, \psi) := F_D(\Omega_t, \varphi) + L_D(\Omega_t, \varphi;\ \psi), \quad \text{where } \varphi \in H^1(\Omega_t) \text{ and } \psi \in H^1_0(\Omega_t),$$

$$G_R(t, \varphi, \psi) := F_R(\Omega_t, \varphi) + L_R(\Omega_t, \varphi;\ \psi), \quad \text{where } \varphi \in H^1(\Omega_t) \text{ and } \psi \in V(\Omega_t).$$

Here[7]

$$F_D(\Omega_t, \varphi) = \frac{1}{2} \int_{\Omega_t} |\nabla \varphi|^2 \, dx_t,$$

[7] We have used here the identity $\int_{\partial\Omega_t} \varphi \partial_{\mathbf{n}} \psi \, ds_t = \int_{\Omega_t} (\varphi \Delta \psi + \nabla \varphi \cdot \nabla \psi) \, dx_t$, where we actually assumed that $\varphi \in H^2(\Omega_t)$ and $\psi \in H^1_0(\Omega_t) \cap H^2(\Omega_t)$. The aforesaid higher regularity of the variables is justified in Step 2 of the proof.

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

$$\begin{aligned}
F_R(\Omega_t, \varphi) &= \frac{1}{2} \int_{\Omega_t} |\nabla \varphi|^2 \, dx_t - \int_\Gamma \partial_{\mathbf{n}} \varphi \, ds, \\
L_D(\Omega_t, \varphi;\ \psi) &= \int_{\Omega_t} (\psi \Delta \varphi + \varphi \Delta \psi + \nabla \varphi \cdot \nabla \psi) \, dx_t - \int_\Gamma \partial_{\mathbf{n}} \psi \, ds, \\
L_R(\Omega_t, \varphi;\ \psi) &= \int_{\Omega_t} \nabla \varphi \cdot \nabla \psi \, dx_t + \int_{\Sigma_t} (\beta \varphi - \lambda) \psi \, ds_t - \int_\Gamma (\varphi - 1) \partial_{\mathbf{n}} \psi \, ds.
\end{aligned}$$

Henceforward, we let

$$\begin{aligned}
V_1(\Omega) &= \{v \in H^1(\Omega) \mid \ v = 1 \text{ on } \Gamma\}, \\
W(\Omega) &= \{v \in H^1(\Omega) \mid \ v = 1 \text{ on } \Gamma \text{ and } v = 0 \text{ on } \Sigma\},
\end{aligned}$$

and use similar notations for functional spaces defined on perturbed domains.

Now, at $t = 0$, we have that

$$J_D(\Omega) = \min_{\varphi \in H^1(\Omega)} \sup_{\psi \in H_0^1(\Omega)} G_D(0, \varphi, \psi); \qquad J_R(\Omega) = \min_{\varphi \in H^1(\Omega)} \sup_{\psi \in V(\Omega)} G_R(0, \varphi, \psi),$$

since, respectively, we have

$$\sup_{\psi \in H_0^1(\Omega)} G_D(0, \varphi, \psi) = \begin{cases} F(\Omega, u_D) & \text{if } \varphi = u_D, \\ +\infty & \text{otherwise;} \end{cases} \tag{14}$$

$$\sup_{\psi \in V(\Omega)} G_R(0, \varphi, \psi) = \begin{cases} F(\Omega, u_R) & \text{if } \varphi = u_R, \\ +\infty & \text{otherwise.} \end{cases} \tag{15}$$

Let us check our claim for $J_R(\Omega)$ (the same argument applies for confirming claim (15)). Indeed, at $t = 0$, we get

$$G_R(0, \varphi, \psi) = F_R(\Omega_0, \varphi) + L_R(\Omega_0, \varphi;\ \psi),$$

where $\varphi \in H^1(\Omega_0)$ and $\psi \in V(\Omega_0)$. Since $\Omega_0 = \Omega$, we can write

$$\begin{aligned}
G_R(0, \varphi, \psi) = \frac{1}{2} \int_\Omega |\nabla \varphi|^2 \, dx - \int_\Gamma \partial_{\mathbf{n}} \varphi \, ds \\
+ \int_\Omega \nabla \varphi \cdot \nabla \psi \, dx + \int_\Sigma (\beta \varphi - \lambda) \psi \, ds - \int_\Gamma (\varphi - 1) \partial_{\mathbf{n}} \psi \, ds,
\end{aligned}$$

for $\varphi \in H^1(\Omega)$ and $\psi \in V(\Omega)$. Taking $\varphi = u_R \in H^1(\Omega)$ gives us

$$\begin{aligned}
G_R(0, \varphi, \psi) = \frac{1}{2} \int_\Omega |\nabla u_R|^2 \, dx - \int_\Gamma \partial_{\mathbf{n}} u_R \, ds \\
+ \int_\Omega \nabla u_R \cdot \nabla \psi \, dx + \int_\Sigma (\beta u_R - \lambda) \psi \, ds - \int_\Gamma (u_R - 1) \partial_{\mathbf{n}} \psi \, ds,
\end{aligned}$$

for $\psi \in V(\Omega)$. The last integral over Γ above vanishes since $u_R = 1$ on Γ. Meanwhile, the third and fourth integrals also vanish. To see this, we multiply the Laplace equation in (4) by any function $\psi \in V(\Omega)$ and apply integration by parts to obtain $a(u_R, \psi) = b(\psi)$ (cf. problem (18) below). Clearly, $G_R(0, \varphi, \psi) = F_R(\Omega, \varphi)$, for $\varphi = u_R$. On the other hand, if $\varphi \neq u_R$, then the supremum of $G_R(0, \varphi, \psi)$ blows-up because it is convex continuous with respect to φ (note that $F_R(\Omega, \varphi)$ is non-negative) and concave continuous with respect to ψ. This verifies assertion (15).

Now, by a similar observation, $G_D(0, \varphi, \psi)$ is also convex continuous with respect to its first argument φ and concave continuous with respect to the second argument ψ. Hence, in accordance with [41, Chap. IV], these functionals both admit a saddle point. Particularly, G_D has the saddle point (u_D, p_D) provided it satisfies the following variational problems:

$$\begin{cases} \text{find } u_D \in W(\Omega) \text{ such that} \\ \qquad\qquad (\nabla u_D, \nabla\psi)_\Omega = 0, \quad \forall\varphi \in H_0^1(\Omega); \end{cases} \tag{16}$$

$$\begin{cases} \text{find } p_D \in V(\Omega),\ p_D = -u_D \text{ on } \Sigma, \text{ such that} \\ \qquad\qquad (\nabla p_D, \nabla\psi)_\Omega = 0, \quad \forall\varphi \in H_0^1(\Omega). \end{cases} \tag{17}$$

On the other hand, G_R has the saddle point (u_R, p_R) provided that it is a solution to the following variational problems:

$$\begin{cases} \text{find } u_R \in V_1(\Omega) \text{ such that} \\ \qquad\qquad a(u_R, \varphi) = b(\varphi), \quad \forall\varphi \in V(\Omega); \end{cases} \tag{18}$$

$$\begin{cases} \text{find } p_R \in V(\Omega) \text{ such that} \\ \qquad\qquad a(p_R, \varphi) = \langle \beta u_R, \varphi\rangle_\Sigma - b(\varphi), \quad \forall\varphi \in V(\Omega). \end{cases} \tag{19}$$

The saddle points (u_D, p_D) and (u_R, p_R) are actually unique due to the unique solvability of the variational problems (16)–(19).[8] Note here that equations (16) and (18) are in fact the respective weak formulations of the state equations (3) and (4).

A similar analysis also holds on the transformed domain Ω_t, and we have

$$\begin{aligned} J_D(\Omega_t) &= \min_{\varphi\in H^1(\Omega_t)} \sup_{\psi\in H_0^1(\Omega_t)} G_D(t, \varphi, \psi), \\ J_R(\Omega_t) &= \min_{\varphi\in H^1(\Omega_t)} \sup_{\psi\in V(\Omega_t)} G_R(t, \varphi, \psi). \end{aligned} \tag{20}$$

The saddle point $(u_{Dt}, p_{Dt}) := (u_D(\Omega_t), p_D(\Omega_t))$ of $G_D(t, \varphi, \psi)$ is characterized by the following system of equations

$$L_D(\Omega_t, u_{Dt};\ \varphi) = 0, \quad \forall\varphi \in H_0^1(\Omega_t), \tag{21}$$

$$dF_D(\Omega_t, u_{Dt};\ \varphi) + dL_D(\Omega_t, u_{Dt};\ p_{Dt};\ \varphi) = 0, \quad \forall\varphi \in H_0^1(\Omega_t). \tag{22}$$

Obviously, at $t = 0$, equations (21) and (22) respectively coincide with (16) and (17). On the other hand, the saddle point $(u_{Rt}, p_{Rt}) := (u_R(\Omega_t), p_R(\Omega_t))$ of $G_R(t, \varphi, \psi)$ is characterized by the set of equations

$$L_R(\Omega_t, u_{Rt};\ \varphi) = 0, \quad \forall\varphi \in V(\Omega_t), \tag{23}$$

$$dF_R(\Omega_t, u_{Rt};\ \varphi) + dL_R(\Omega_t, u_{Rt};\ p_{Rt};\ \varphi) = 0, \quad \forall\varphi \in V(\Omega_t), \tag{24}$$

which, when $t = 0$, are equivalent to problems (18)–(19).

Step 2. Our goal is to get the derivative of the functionals $G_D(t, \varphi, \psi)$ and $G_R(t, \varphi, \psi)$ with respect to the parameter $t \geqslant 0$ through the application of Theorem A.1 furnished in [43]. The application of the said theorem, however, is not actually straightforward because the function spaces appearing in (20) also depend on t. To circumvent this issue, we employ the so-called function space embedding technique put forward in [35, Chap. 10.6].[9]

[8] In a slightly more general case, the existence of (unique) weak solution to the variational form of the Robin problem (4) also follows from Lax-Milgram lemma provided, in particular, that the Robin term $\beta := \beta(x) \in L^\infty(\Sigma)$, and is positive almost everywhere in the free boundary (cf., e.g, [42, Lem. 7.36.3, p. 617]).

[9] In this method, a large enough set that contains all admissible transformations of Ω is introduced, making the Lagrangians involved in the formulation to admit non-singleton sets of saddle points.

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

Before we proceed, let us discuss shortly the regularity of the saddle points. For a domain Ω of class $C^{k,1}$, $k \geqslant 1$, the saddle point (u_D, p_D) belongs to $[H^{k+1}(\Omega)]^2$ (see, e.g., [16, Thm. 29]). Analogously, the saddle point (u_R, p_R) is also an element of $[H^{k+1}(\Omega)]^2$ whenever Ω is of class $C^{k,1}$ (see, e.g., [44, Rem. 3.5]).[10] Furthermore, if $\boldsymbol{V} \in \boldsymbol{\Theta}^k$, then we can also say that the pairs (u_{Dt}, p_{Dt}) and (u_{Rt}, p_{Rt}) both belong to $[H^{k+1}(\Omega_t)]^2$ since T_t is a $C^{k,1}$ diffeomorphism.

Now, by our initial assumption on Ω and $\boldsymbol{V}$, we can get the H^2 regularity of the saddle points. Because $\mathbb{R}^d \supset \overline{U} \supset \{\overline{\Omega}_t \mid t \in \mathcal{I}\}$, then we can write

$$J_i(\Omega_t) = \min_{\Phi \in H^2(\mathbb{R}^d)} \sup_{\Psi \in H^2(\mathbb{R}^d)} G_i(t, \Phi, \Psi), \tag{25}$$

for each index $i \in \{D, R\}$, where the indices D and R correspond to every expressions or terms associated with the pure Dirichlet problem and mixed Dirichlet-Robin problem, and to the Lagrangians G_D and G_R, respectively.[11] In above min-sup expression, the functional $G_i(t, \Phi, \Psi)$ is defined as

$$G_i(t, \Phi, \Psi) = F(\Omega_t, \Phi) + L_i(\Omega_t, \Phi, \Psi), \tag{26}$$

where $(\Phi, \Psi)|_{\Omega_t} = (u_{it}, p_{it})$, for each $i \in \{D, R\}$.

The set of saddle points of $G_i(t, \Phi, \Psi)$, $i \in \{D, R\}$, which we denote here by $S_i(t)$, are not singletons since $S_i(t) = X_i(t) \times Y_i(t) \subset [H^2(\mathbb{R}^d)]^2$, where

$$X_i(t) = \{\Phi \in H^2(\mathbb{R}^d) \mid \Phi|_{\Omega_t} = u_{it}\} \quad \text{and} \quad Y_i(t) = \{\Psi \in H^2(\mathbb{R}^d) \mid \Psi|_{\Omega_t} = p_{it}\},$$

for each $i \in \{D, R\}$. Here, the pair $(u_{Dt}, p_{Dt}) \in [H^2(\Omega_t)]^2 \cap [W(\Omega_t) \times H_0^1(\Omega_t)]$ is the unique solution to the system of equations (21)–(22), and $(u_{Rt}, p_{Rt}) \in [H^2(\Omega_t)]^2 \cap [V_1(\Omega_t) \times V(\Omega_t)]$, on the other hand, solves the other system of equations (23)–(24).

Step 3. Our next objective is to find an expression for the limit given in (8). For this purpose, we apply Theorem A.1 so to obtain the shape gradient of the given cost functional J. Under appropriate assumptions (to be verified in the next step), Theorem A.1 states that

$$dJ(\Omega)[\boldsymbol{V}] = \min_{\Phi \in X_D(0)} \sup_{\Psi \in Y_D(0)} \partial_t G_D(t, \Phi, \Psi) + \min_{\Phi \in X_R(0)} \sup_{\Psi \in Y_R(0)} \partial_t G_R(t, \Phi, \Psi). \tag{27}$$

Since the sets $S_D(0)$ and $S_R(0)$ have been identified, we only need to compute the partial derivative of the Lagrangians $G_i(t, \Phi, \Psi)$, $i \in \{D, R\}$ with respect to t, given in (26). To this end, we now assume that Ω is of class $C^{2,1}$ and $\boldsymbol{V} \in \boldsymbol{\Theta}^2$. So, we can choose to consider the saddle points to be $[H^3(\mathbb{R}^d)]^2$ regular. This higher regularity is crucial, especially for the integrand having the Laplacian term. Because the saddle points have sufficient regularities, we can apply Hadamard's domain and boundary differentiation formulas (see [35, Thm. 4.2, p. 483] and [35, Thm. 4.3, p. 486], respectively, see also [45, 46]):

$$\left.\frac{d}{dt} \int_{\Omega_t} f(t, x)\, dx_t \right|_{t=0} = \int_{\Omega} \partial_t f(0, x)\, dx + \int_{\partial\Omega} f(0, \sigma) V_n \, ds, \tag{28}$$

$$\left.\frac{d}{dt} \int_{\partial\Omega_t} f(t, \sigma)\, ds_t \right|_{t=0} = \int_{\partial\Omega} \partial_t f(0, \sigma)\, ds + \int_{\partial\Omega} \left(\partial_{\mathbf{n}} f(0, \sigma) + \kappa f(0, \sigma)\right) V_n \, ds, \tag{29}$$

to obtain – noting that $\boldsymbol{V} = \mathbf{0}$ on Γ – the partial derivative

$$\partial_t G_i(t, \Phi, \Psi) = \int_{\Sigma_t} \mathcal{G}_i(\Phi, \Psi) \boldsymbol{V} \cdot \mathbf{n}_t \, ds_t, \qquad i \in \{D, R\},$$

[10] For more details about existence and uniqueness of solutions to mixed Robin-Dirichlet problems in $W^{s,2}$ for bounded domains in $\mathbb{R}^d$, $d \in \{2, 3\}$, one may consult [42, Sec. 7.36].

[11] From this point onwards, we occasionally use this notation to shorten some statements, and for brevity.

where

$$\mathcal{G}_D(\Phi, \Psi) = \frac{1}{2}|\nabla\Phi|^2 + \Psi\Delta\Phi + \Phi\Delta\Psi + \nabla\Phi \cdot \nabla\Psi,$$
$$\mathcal{G}_R(\Phi, \Psi) = \frac{1}{2}|\nabla\Phi|^2 + \nabla\Psi \cdot \nabla\Phi + \partial_{\mathbf{n}}((\beta\Phi - \lambda)\Psi) + \kappa_t(\beta\Phi - \lambda)\Psi,$$

and κ_t denotes the mean curvature of Σ_t while $\mathbf{n}_t$ is the outward unit normal to Σ_t. Notice from above derivative that the integral is defined only on the boundary Σ_t which will not depend on Φ and Ψ outside of $\overline{\Omega}_t$. Since the restriction of the elements of $S_D(0)$ and $S_R(0)$ are unique, then we can actually drop the $\min$ and $\sup$ in (27). Doing so leads us to the following expression for the shape derivative of J

$$\begin{aligned} dJ(\Omega)[V] = &\int_\Sigma \frac{1}{2}\left((\partial_{\mathbf{n}}u_R)^2 - (\partial_{\mathbf{n}}u_D)^2 + (\partial_\tau u_R)^2\right) V_n \, ds \\ &+ \int_\Sigma (\nabla_\Sigma u_R \cdot \nabla_\Sigma p_R + \partial_{\mathbf{n}}u_R \partial_{\mathbf{n}}p_R) \, V_n \, ds \\ &+ \int_\Sigma (\beta\partial_{\mathbf{n}}u_R p_R + (\beta u_R - \lambda)(\partial_{\mathbf{n}}p_R + \kappa p_R)) \, V_n \, ds. \end{aligned}$$

In above, we have used the fact that $p_D = -u_D$ and $u_D = 0$ on Σ. The former equation implies that $\partial_{\mathbf{n}}p_D = -\partial_{\mathbf{n}}u_D$ on Σ while the latter implies that $|\nabla u_D|^2 = (\partial_{\mathbf{n}}u_D)^2$ on Σ. Here, the operator ∇_Σ stands for the tangential gradient (see, e.g., [35, Chap. 9, Sec. 5.2, eq. (5.17), p. 497]) on Σ defined as $\nabla_\Sigma(\cdot) = \nabla(\cdot)|_\Sigma - [\nabla(\cdot) \cdot \mathbf{n}]\mathbf{n}$.

In summary, we obtain the characterization of the shape derivative of J given by $dJ(\Omega)[V] = \int_\Sigma \mathcal{G}_\beta \mathbf{n} \cdot V \, ds$, where the kernel $\mathcal{G}_\beta$ is specified by (9). Notably, the computed expression for $dJ(\Omega)[V]$ accords with the Hadamard-Zolésio structure theorem (see, e.g., [35, Thm. 3.6, p. 479]); that is, there exists – for a $C^{2,1}$ domain Ω – a scalar distribution $\mathcal{G}_\beta \in \mathcal{D}^1(\Sigma)$ such that $dJ(\Sigma)[V] = \langle \mathcal{G}_\beta, V_n \rangle$.

Step 4. To finish the proof of the proposition, the four assumptions of Theorem A.1 have to be verified. We only sketch the argumentations since they are similar to [35, Chap. 10, Sec. 6.4, pp. 566–570].

The first hypothesis (H1) is easily verified by the use of linear and continuous extensions $\Pi : H^k(\Omega) \to H^k(\mathbb{R}^d)$ and $\Pi_t : H^k(\Omega_t) \to H^k(\mathbb{R}^d)$, $k \in \mathbb{N}$, (see, e.g., [47, p. 146]) from which we can get the existence of elements of $S_D(t)$ and $S_R(t)$.

Meanwhile, the second hypothesis (H2) is checked without difficulty by utilizing – after transforming the integrals over the reference domain – the regularities of the maps $t \mapsto I_t$, $t \mapsto A_t$, and $t \mapsto B_t$ given in (11) and together with the H^3 regularity of the state and adjoint states.

To check conditions (H3)(i) and (H4)(i), we put into use the bounds in (13) to first show that $u_D^t = u_{Dt} \circ T_t$, $u_R^t = u_{Rt} \circ T_t$, $p_D^t = p_{Dt} \circ T_t$, and $p_R^t = p_{Rt} \circ T_t$ – the transported solutions from Ω_t to Ω of equations (21)–(24) – are bounded in $H^1(\Omega)$. Since $H^1(\Omega)$ is a Hilbert space, we can extract weakly convergent subsequences converging to some points $(u_D^0, p_D^0), (u_R^0, p_R^0) \in [H^1(\Omega)]^2$. However, the limit points (u_D^0, p_D^0) and (u_R^0, p_R^0) will coincide with the solutions (u_D, p_D) and (u_R, p_R) of (16)–(17) and (18)–(19), respectively. This follows from the linearity of the transported versions of equations (21)–(24) with respect to (u_D^t, p_D^t) and (u_R^t, p_R^t), the continuity of I_t, A_t, and B_t, and the fact that the aforementioned systems have unique solutions $[H^1(\Omega)]^2$.

Next, we need to exhibit that the weak convergences in $H^1(\Omega)$ are actually strong. To do this we estimate the norms $\|u_D^t - u_D\|_{H^1(\Omega)}$, $\|u_R^t - u_R\|_{H^1(\Omega)}$, $\|p_D^t - p_D\|_{H^1(\Omega)}$, and $\|p_R^t - p_R\|_{H^1(\Omega)}$ and then deduce the $H^1(\Omega)$-strong convergences $u_D^t \to u_D$, $u_R^t \to u_R$,

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

$p_D^t \to p_D$, and $p_R^t \to p_R$ using the fact that $I_t \to 1$, $A_t - id \to 0$, and $B_t \to 1$, as $t \to 0$. The details of the computation is standard so we skip them.

Now, given that $\Omega \in C^{2,1}$ and $\boldsymbol{V} \in \boldsymbol{\Theta}^2$, we can also show by means of classical regularity theorems [48] that u_D^t, u_R^t, p_D^t, and p_R^t are also bounded in $H^3(\Omega)$ since these functions also live in the space $H^3(\Omega)$. Therefore, we also have the continuity of the functions in $H^3(\Omega)$.

In light of [35, Thm. 6.2, p. 568], we see that the convergences $\Pi_t u_{Dt} \to \Pi u_D$, $\Pi_t u_{Rt} \to \Pi u_R$, $\Pi_t p_{Dt} \to \Pi p_D$, and $\Pi_t p_{Rt} \to \Pi p_R$ all hold strongly in $H^3(\mathbb{R}^d)$. Thus, assumptions (H3)(i) and (H4)(i) are satisfied for the $H^3(\mathbb{R}^d)$-strong topology.

Lastly, to verify conditions (H3)(ii) and (H4)(ii), we write

$$\partial_t G_i(t, \Phi, \Psi) = \int_{\Omega_t} \operatorname{div}\left(\mathcal{G}_i(\Phi, \Psi)\boldsymbol{V}\right) dx_t =: \int_{\Omega_t} \operatorname{div}\left(\mathcal{F}_i(\Phi, \Psi)\right) dx_t,$$

for each $i \in \{D, R\}$. Clearly, the map

$$(\Phi, \Psi) \longmapsto \mathcal{F}_i(\Phi, \Psi) : [H^3(\mathbb{R}^d)]^2 \longrightarrow H^1(\mathbb{R}^d)$$

is bilinear and continuous, for each $i \in \{D, R\}$. In addition, the mapping

$$(t, \mathcal{F}_i) \longmapsto \int_{\Sigma_t} \mathcal{F}_i \cdot \mathbf{n}_t \, ds_t = \int_{\Omega} \operatorname{div}\left(\mathcal{F}_i\right) \circ T_t I_t \, dx,$$

is also continuous from $\mathcal{I} \times H^1(\mathbb{R}^d)$ to $\mathbb{R}$, for each $i \in \{D, R\}$. Finally, the map

$$(t, \Phi, \Psi) \mapsto \partial_t G_i(t, \Phi, \Psi) = \int_{\Sigma_t} \mathcal{F}_i(\Phi, \Psi) \cdot \mathbf{n}_t \, ds_t$$

is continuous, for each $i \in \{D, R\}$. Thus, conditions (H3)(ii) and (H4)(ii) are also verified. This completes the verification of the assumptions of Theorem A.1 proving that equation (27) holds true.

The last statement of the proposition can easily be inferred from (9). Indeed, if $\Omega = \Omega^*$, where Ω^* solves (1), then it holds that $u = u_D = u_R$ on $\overline{\Omega}^*$. Hence, $u = 0$ and $\partial_{\mathbf{n}} u = \lambda$ on Σ_*. This means, in particular, that $\partial_\tau u_R = 0$, $\nabla u_R = \partial_{\mathbf{n}} u_R \mathbf{n}$, and $(\partial_{\mathbf{n}} u_R)^2 = \lambda^2$ on Σ_*. It therefore follows that $\mathcal{G}_\beta \equiv 0$ which in turn implies that $dJ(\Omega)[\boldsymbol{V}] = 0$ at $\Omega = \Omega^*$. This completes the proof of Proposition 2.1.1. □

Remark 1 For $\beta \equiv 0$, the shape gradient reduces to

$$\mathcal{G}_0 = -\frac{1}{2}(\partial_{\mathbf{n}} u_D)^2 + \partial_\tau u_R \partial_\tau p_R - \lambda \kappa p_R + \frac{1}{2}\lambda^2 + \frac{1}{2}(\partial_\tau u_R)^2.$$

Note that the adjoint p_R can actually be expressed in terms of the state u_R on Σ. In fact, we have $p_R = -u_R$ on Σ. Indeed, from (24) with $t = 0$ (i.e., the case where integration is carried out over the reference domain Ω), one easily finds that $\int_\Omega \varphi \Delta(p_R + u_R)\, dx = 0$, for all $\varphi \in C_0^\infty(\Omega)$, with $\partial_{\mathbf{n}} \varphi = 0$ on Γ. Varying φ, the equation leads to $-\Delta p_R = \Delta u_R = 0$ in Ω. Now, let us choose that $\varphi \in H^2(\Omega) \cap V(\Omega)$ with $\Delta \varphi = 0$ in Ω. So, again from (24) with $t = 0$, and by Green's identity, we have $\int_\Omega \nabla(p_R + u_R) \cdot \nabla \varphi \, dx - \int_\Gamma \partial_{\mathbf{n}} \varphi \, ds = \int_\Gamma p_R \partial_{\mathbf{n}} \varphi \, ds + \int_\Sigma (p_R + u_R) \partial_{\mathbf{n}} \varphi \, ds = 0$, for any $\varphi \in H^2(\Omega) \cap V(\Omega)$ such that $\Delta \varphi = 0$ in Ω. Varying the normal trace $\partial_{\mathbf{n}} \varphi$ on Γ, we find that $p_R = 0$ on Γ. Similarly, varying the trace of the normal derivative $\partial_{\mathbf{n}} \varphi$ on Σ, we get $p_R = -u_R$ on Σ. Hence, we may actually write $\mathcal{G}_0$ above as

$$\mathcal{G}_0 = \frac{1}{2}\left(\lambda^2 - (\partial_{\mathbf{n}} u_D)^2 + 2\lambda \kappa u_R - (\partial_\tau u_R)^2\right), \tag{30}$$

which is exactly the shape gradient of the Kohn-Vogelius cost functional

$$J_0(\Omega) := \frac{1}{2} \int_{\Omega} |\nabla(u_D - u_{\mathrm{N}})|^2 \, dx,$$

where $u_{\mathrm{N}} = u_R$ is now the solution to the mixed Dirichlet-Neumann problem (4) with $\beta \equiv 0$ (cf. [16, Thm. 33, eq. (142)], [14, Thm. 2, eq. (19)], [18, Thm. 1, eq. (4)]).

In the conclusion drawn in [16], the authors stated that the strategy – sometimes referred to as the rearrangement method – they used to compute the shape gradient of J_0 does not require the introduction of the adjoint variables. This is contrary to the computation of the shape gradient (derived using the same method) for the Dirichlet- and Neumann-data tracking functionals presented respectively in [9] and [11]. Although it was not stated in [16], it should be apparent here that the very reason why the computation presented in [16] did not require the used of adjoint variables is that the Kohn-Vogelius functional with the associated states u_D and u_{N} (where u_{N} denotes the solution of (4) when $\beta = 0$) is actually self-adjoint. This means, in a sense, that the corresponding adjoint variables to u_D and u_{N} can actually be expressed in terms of the latter states alone.

Remark 2 Note that when $\beta = \kappa$ where κ the mean curvature of Σ, the shape gradient $\mathcal{G}_\beta$ given by (9) becomes

$$\mathcal{G}_\kappa := \frac{1}{2} \left((\partial_{\mathbf{n}} u_R)^2 - (\partial_{\mathbf{n}} u_D)^2 + 2 \partial_\tau u_R \partial_\tau p_R + (\partial_\tau u_R)^2 \right). \tag{31}$$

Remark 3 Recall in Step 3 of the proof that we required Ω to be of class $C^{2,1}$ and choosed $\boldsymbol{V} \in \boldsymbol{\Theta}^2$ in order to apply Hadamard's domain and boundary differentiation formulas. These regularity assumptions can be relaxed if we apply the rearrangement method (see [36]) to get the shape derivative of J. In this case, we only need the reference domain as well as its perturbations be of class $C^{1,1}$. We have only used the minimax formulation in order to demonstrate that J – associated to the case where $\beta = 0$ – is self-adjoint. Moreover, we note that the computed shape derivative of the cost already make sense for Ω of class $C^{1,1}$. In fact, if $\Omega \in C^{1,1}$, then κ is well defined almost everywhere on Σ, and actually belongs to L^∞ (for a proof of this statement, see [49]). Thus, we infer that $\mathcal{G}_\beta$ is not just a distribution, but is also in $L^2(\Sigma)$.

Remark 4 Another rigorous way to exhibit the boundary integral form of dJ – requiring only the mild $C^{1,1}$ regularity for Ω and on $\boldsymbol{V}$ – is to use the variational equations satisfied by the Eulerian derivative of u_D and u_R. This is as opposed to the approach used in [33] which not only requires the shape derivative of the states, but also uses their strong forms in order to write dJ in terms of a boundary integral – that is, in accordance with the Hadamard-Zolésio structure theorem.

We note that the chain rule approach actually provides a shorter more elegant way of obtaining the shape derivative of the cost. However, applying the method means requiring more regularity of the states; see, for example, [45, Thm 5.5.2].

2.2 The Eulerian derivative approach

In this subsection, we formally show that the maps $t \mapsto u_D^t$ and $t \mapsto u_R^t$ are C^1 in a neighborhood of 0 (Proposition 2.2.1), and characterize their Eulerian

Paper accepted to Numerical Algorithms (DOI: 10.1007/s11075-023-01497-x.)

derivatives at 0 in the direction of the field $\boldsymbol{V}$. For this purpose, we fix $\boldsymbol{V} \in \boldsymbol{\Theta}^1$. Afterwards, we prove the same for the map $t \mapsto J(\Omega_t)$ (see Proposition 2.2.5).

Proposition 2.2.1 The mappings $t \mapsto u_D^t \in H^1(\Omega)$ and $t \mapsto u_R^t \in H^1(\Omega)$ are C^1 in a neighborhood of 0. Moreover, their respective Eulerian derivatives at 0, denoted by $\widetilde{u_D}$ and $\widetilde{u_R}$, satisfy the properties that $\widetilde{u_D} \in H_0^1(\Omega)$, $\widetilde{u_R} \in V(\Omega)$, and are respectively solutions of the following variational problems:

$$(\nabla\widetilde{u_D}, \nabla v)_\Omega = -(A\nabla u_D, \nabla v)_\Omega, \quad \forall v \in H_0^1(\Omega), \tag{32}$$

$$\begin{aligned} a(\widetilde{u_R}, v) = -(A\nabla u_R, \nabla v)_\Omega - \int_\Sigma \beta(\operatorname{div}_\Sigma \boldsymbol{V}) u_R v \, ds \\ + \int_\Sigma \lambda(\operatorname{div}_\Sigma \boldsymbol{V}) v \, ds, \quad \forall v \in V(\Omega). \end{aligned} \tag{33}$$

Lemma 2.2.2 The bilinear form a_t defined on $V(\Omega) \times V(\Omega)$ is bounded and coercive on $V(\Omega) \times V(\Omega)$ for $t \in \mathcal{I}$.

Proof This is shown easily using the continuity of A_t and B_t for $t \in \mathcal{I}$. □

Lemma 2.2.3 The functions u_D^t and u_R^t are respectively the unique solutions in $H^1(\Omega)$ of the following equations

$$u_D^t = 1 \text{ on } \Gamma, \quad u_D^t = 0 \text{ on } \Sigma, \quad (A_t \nabla u_D^t, \nabla v)_\Omega = 0, \ \forall v \in H_0^1(\Omega), \tag{34}$$

$$u_R^t = 1 \text{ on } \Gamma, \quad a^t(u_R^t, v) = b(v), \ \forall v \in V(\Omega). \tag{35}$$

Proof We only prove the latter because the first one can be shown in a similar manner. The function $u_{Rt} \in H^1(\Omega_t)$ is the solution of the variational problem

$$u_{Rt} = 1 \text{ on } \Gamma, \quad \int_{\Omega_t} \nabla u_{Rt} \cdot \nabla \varphi_t \, dx_t + \int_{\Sigma_t} \beta u_{Rt} \varphi_t \, ds_t = \int_{\Sigma_t} \lambda \varphi_t \, ds_t, \quad \forall \varphi_t \in V(\Omega_t),$$

where $V(\Omega_t) = \{\varphi_t \in H^1(\Omega_t) \mid \varphi_t = 0 \text{ on } \Gamma\}$. Using the relation $u_R^t = u_{Rt} \circ T_t$, the identity $(\nabla\varphi_t) \circ T_t = DT_t^{-\top} \nabla\varphi^t$, which holds for any $\varphi_t \in H^1(\Omega_t)$ and $\varphi^t \in H^1(\Omega)$, and the change of variables (see [35, subsec. 9.4.2–9.4.3, pp. 482–484]), the variational equation above transforms into

$$a^t(u_R^t, \varphi^t) = b^t(\varphi^t), \quad \forall \varphi^t \in V(\Omega),$$

and $u_R^t = 1$ on Γ. Hence, taking $v = \varphi^t$ we get (35). As shown previously, the bilinear form $a^t(\cdot,\cdot) : V(\Omega) \times V(\Omega) \to \mathbb{R}$ is bounded and coercive. We let $u_0 \in H^1(\mathbb{R}^d)$ be a fixed function such that $u_0 = 1$ on Γ. Clearly, $z^t := u_R^t - u_0 \in V(\Omega)$, and so, by (35), we have

$$a^t(z^t, v) = -a^t(u_0, v) + b^t(v), \quad \forall v \in V(\Omega). \tag{36}$$

For $t \in \mathcal{I}$, it can be verified that the following estimates hold

$$|-a^t(u_0, v) + b^t(v)| \lesssim \|v\|_{H^1(\Omega)}.$$

Hence, by Lax-Milgram lemma, $z^t \in V(\Omega)$ uniquely solves (36).

Now, let $u_R^t = z^t + u_0 \in H^1(\Omega)$. Then, we have

$$a^t(u_R^t, v) = a^t(z^t + u_0, v) = b^t(v), \quad \forall v \in V(\Omega).$$

Because $z^t \in V(\Omega)$, $u_R^t = z^t + 1 = 1$ on Γ. Uniqueness of u_R^t follows from the uniqueness of z^t. Therefore, u_R^t is the unique solution of (35) in $H^1(\Omega)$. □

Lemma 2.2.4 The maps $t \mapsto u_D^t$ and $t \mapsto u_R^t$ are C^1 in a neighborhood of 0.

Proof We prove using the implicit function theorem (IFT) the result for the map $t \mapsto u_R^t$, the proof for $t \mapsto u_D^t$ being similar. Using (35), we note that $z_R^t := u_R^t - u_R$ is the unique element in $V(\Omega)$ that satisfies

$$a^t(z_R^t, v) = -a^t(u_R, v) + b^t(v), \quad \forall v \in V(\Omega).$$

Denoting by $\langle \cdot, \cdot \rangle_V$ the duality pairing between $V(\Omega)$ and its dual space $V'(\Omega)$, we consider the function $\Phi : \mathcal{I} \times V(\Omega) \to V'(\Omega)$ defined by

$$\langle \Phi(t, w), v \rangle_V := a^t(w + u_R, v) - b^t(v),$$

for all $v, w \in V(\Omega)$. Then, we have that z_R^t the unique element in $V(\Omega)$ such that $\Phi(t, z_R^t) = 0$. Note that Φ is C^1 because of (11). Moreover, we have

$$\langle D_w \Phi(0, 0) w, v \rangle_V = a^t(w, v).$$

As was shown earlier, the right side expression above is bounded and coercive on $V(\Omega)$. By the Lax-Milgram lemma, we deduce that $D_w\Phi(0, 0)$ is an isomorphism from $V(\Omega)$ to $V'(\Omega)$. Thus, by virtue of IFT, we conclude that the mapping $t \mapsto z_R^t$ is C^1 in a neighborhood of 0. Now we let $\widetilde{u_R} \in V(\Omega)$ be its derivative at $t = 0$. Differentiating the identity $\Phi(t, z_R^t) = 0$ with respect to t we get

$$\langle D_w \Phi(0, 0)\widetilde{u_R}, v \rangle + \langle \partial_t \Phi(0, 0), v \rangle = 0, \quad \forall v \in V(\Omega).$$

This yields equation (33). □

Next, we prove the differentiability of the map $t \mapsto J(\Omega_t)$ and characterize its derivative.

Proposition 2.2.5 The mapping $t \mapsto J(\Omega_t)$ is C^1 in a neighborhood of 0, and its derivative at 0 is given by $dJ(\Omega)[\boldsymbol{V}] = \int_\Sigma \widetilde{\mathcal{G}}_\beta \mathbf{n} \cdot \boldsymbol{V}\, ds$ where $\widetilde{\mathcal{G}}_\beta$ is given by

$$\begin{aligned} \widetilde{\mathcal{G}}_\beta = \frac{1}{2}\Big(&(\partial_{\mathbf{n}} u_R)^2 - (\partial_{\mathbf{n}} u_D)^2 + 2\lambda\kappa u_R - (\partial_\tau u_R)^2 \Big) \\ &+ \left(\nabla_\Sigma q_R \cdot \nabla_\Sigma u_R - \lambda\kappa q_R + \beta\left(\partial_{\mathbf{n}} u_R + \kappa u_R\right)\left(q_R - u_R\right) \right), \end{aligned} \tag{37}$$

where the adjoint variable $q_R \in V(\Omega)$ satisfies

$$a(q_R, \varphi) = \int_\Sigma \beta u_R \varphi \, ds, \quad \forall \varphi \in V(\Omega), \tag{38}$$

and κ is the mean curvature of Σ.

Remark 5 Observe from (38) that when $\beta = 0$, $q_R \equiv 0$ on $\overline{\Omega}$. Moreover, $\widetilde{\mathcal{G}}_\beta$ reduces to $\mathcal{G}_0$. That is, we obtain [16, Thm. 33, eq. (142)] (see also [14, Thm. 2, eq. (19)], [18, Thm. 1, eq. (4)]) as a corollary of Proposition 2.2.5. Meanwhile, if we take $\beta = \kappa$, then $\partial_{\mathbf{n}} u_R + \kappa u_R = \lambda$ on Σ, and $\widetilde{\mathcal{G}}_\beta$ becomes

$$\widetilde{\mathcal{G}}_\kappa = \frac{1}{2}\left((\partial_{\mathbf{n}} u_R)^2 - (\partial_{\mathbf{n}} u_D)^2 + 2\nabla_\Sigma q_R \cdot \nabla_\Sigma u_R - (\partial_\tau u_R)^2\right). \tag{39}$$

Moreover, at the shape solution Ω^* of the exterior Bernoulli problem (1), we have that $\widetilde{\mathcal{G}}_\beta = 0$ on Σ_*, regardless of the choice of β.

To prove Proposition 2.2.5, we need the following lemma. The $C^{1,1}$ regularity of Ω which implies that $u_D, u_R \in H^2(\Omega)$ will be used subsequently without further notice.

Lemma 2.2.6 Let $\boldsymbol{V} \in \boldsymbol{\Theta}^1$. Then, the solutions u_D and u_R of problems (3) and (4) respectively satisfy the equations

$$(A\nabla u_D, \nabla u_D)_\Omega = -\int_\Sigma (\nabla u_D \cdot \mathbf{n})^2 V_n \, ds, \tag{40}$$

$$(A\nabla u_R, \nabla u_R)_\Omega = \int_\Sigma |\nabla u_R|^2 V_n \, ds + 2\int_\Sigma (\beta u_R - \lambda)(\boldsymbol{V} \cdot \nabla u_R) \, ds. \tag{41}$$

Proof Let us first note that for bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, the identity

$$-\int_\Omega (\operatorname{div} \mathbf{v})\phi \, dx = \int_\Omega \mathbf{v} \cdot \nabla\phi \, dx - \int_\Sigma \phi(\mathbf{v} \cdot \mathbf{n}) \, ds,$$

holds for any vector field $\mathbf{v} \in C^1(\overline{\Omega})^d$ and scalar function $\phi \in W^{1,1}(\Omega)$. Recalling that $A = (\operatorname{div} \boldsymbol{V}) id - D\boldsymbol{V} - (D\boldsymbol{V})^\top$ (see (12)), we note that for harmonic functions $u, v \in H^2(\Omega)$ and vector field $\boldsymbol{V} \in \boldsymbol{\Theta}^1$ we have

$$-(A\nabla v, \nabla u)_\Omega = \int_\Sigma (\partial_{\mathbf{n}} v(\boldsymbol{V} \cdot \nabla u) + \partial_{\mathbf{n}} u(\boldsymbol{V} \cdot \nabla v) - (\nabla v \cdot \nabla u)V_n) \, ds; \tag{42}$$

see, e.g., the proof of Lemma 32 in [16] or the proof of Lemma 5 in [14].

Because $\nabla u_D \cdot \nabla u_R \in W^{1,1}(\Omega)$, $\nabla u_D = (\nabla u_D \cdot \mathbf{n})\,\mathbf{n}$, $\nabla u_R \cdot \mathbf{n} = -\beta u_R + \lambda$, and $-\Delta u_D = -\Delta u_R = 0$, we get (40) and (41) from the above formulas with $u = v = u_D$ and $u = v = u_R$, respectively. □

Proof of Proposition 2.2.5 In the proof, for notational convenience, we write

$$w^t := u_D^t - u_R^t \qquad \text{and} \qquad w := u_D - u_R.$$

Now, by change of variables, we have

$$J(\Omega_t) = \frac{1}{2}\int_\Omega A_t \nabla w^t \cdot \nabla w^t \, dx.$$

Using (11) and Lemma 2.2.4, we deduce that the map $t \mapsto J(\Omega_t)$ is also C^1 in a neighborhood of 0. Then, we can differentiate $J(\Omega_t)$ above with respect to t and get

$$dJ(\Omega)[\boldsymbol{V}] = \frac{1}{2}\int_\Omega A\nabla w \cdot \nabla w \, dx + \frac{1}{2}\int_\Omega \nabla w \cdot (\nabla \widetilde{u_D} - \nabla \widetilde{u_R}) \, dx.$$

Let us write the second integral above in terms of the state variables alone. First, we take $\varphi = \widetilde{u_D} \in H_0^1(\Omega)$ in (16) and (18) to get

$$(\nabla u_D, \nabla \widetilde{u_D})_\Omega = 0, \quad \text{and} \quad a(u_R, \widetilde{u_D}) = b(\widetilde{u_D}).$$

Next, we take $v = u_D - u_R \in V(\Omega)$ in (33), and note that $u_D = 0$ on Σ, to obtain

$$a(\widetilde{u_R}, w) = -\int_\Omega A\nabla u_R \cdot \nabla w \, dx + \int_\Sigma \beta \operatorname{div}_\Sigma \boldsymbol{V} |\nabla u_R|^2 \, ds - \int_\Sigma \lambda (\operatorname{div}_\Sigma \boldsymbol{V}) u_R \, ds.$$

These computations lead to

$$\begin{aligned} dJ(\Omega)[\boldsymbol{V}] = \frac{1}{2}\int_\Omega &(A\nabla u_D \cdot \nabla u_D - A\nabla u_R \cdot \nabla u_R)\, dx \\ &- \int_\Sigma \beta \operatorname{div}_\Sigma \boldsymbol{V} |\nabla u_R|^2 \, ds + \int_\Sigma \lambda (\operatorname{div}_\Sigma \boldsymbol{V}) u_R \, ds - \int_\Sigma \beta u_R \widetilde{u_R} \, ds. \end{aligned} \tag{43}$$

The above representation of the shape derivative of J in the direction of $\boldsymbol{V}$ at Ω is actually not useful for practical applications, especially in the numerical realization of the minimization problem (5). The main difficulty arises from the fact that (33) has to be solved for each velocity field $\boldsymbol{V}$ in order to evaluate $dJ(\Omega)[\boldsymbol{V}]$. A way to resolve this issue is to apply the adjoint method. To do this, we introduce the adjoint variable $q_R \in V(\Omega)$ satisfying the variational problem given in (38). Then, we take $\varphi = \widetilde{u_R} \in V(\Omega)$ in (38) and let $v = q_R \in V(\Omega)$ in (33) of Proposition 2.2.1 to obtain

$$\int_\Sigma \beta u_R \widetilde{u_R} \, ds = -\int_\Omega A\nabla u_R \cdot \nabla q_R \, dx - \int_\Sigma \beta (\operatorname{div}_\Sigma \boldsymbol{V}) u_R q_R \, ds + \int_\Sigma \lambda (\operatorname{div}_\Sigma \boldsymbol{V}) q_R \, ds.$$

This gives us the expression

$$\begin{aligned} dJ(\Omega)[\boldsymbol{V}] = \frac{1}{2}\int_\Omega &(A\nabla u_D \cdot \nabla u_D - A\nabla u_R \cdot \nabla u_R)\, dx \\ &- \int_\Sigma \beta |\nabla u_R|^2 \operatorname{div}_\Sigma \boldsymbol{V} \, ds + \int_\Sigma \lambda u_R \operatorname{div}_\Sigma \boldsymbol{V} \, ds \\ &+ \int_\Omega A\nabla u_R \cdot \nabla q_R \, dx + \int_\Sigma \beta u_R q_R \operatorname{div}_\Sigma \boldsymbol{V} \, ds - \int_\Sigma \lambda q_R \operatorname{div}_\Sigma \boldsymbol{V} \, ds. \end{aligned} \tag{44}$$

The above integral already make sense, but we opt to write it in terms of a boundary integral form with the help of Lemma 2.2.6 and through the application of following version of the tangential Green's formula[12], which is valid when Σ is $C^{1,1}$,

$$\int_\Sigma (\nabla \phi \cdot \boldsymbol{V} + \phi \operatorname{div}_\Sigma \boldsymbol{V})\, ds = \int_\Sigma (\partial_{\mathbf{n}} \phi + \phi \operatorname{div}_\Sigma \mathbf{n})\, V_n \, ds. \tag{45}$$

The function ϕ here is $W^{2,1}$ regular in the hold-all domain U. To proceed, we first apply Lemma 2.2.6 so that, after a few rearrangements, we get

$$\begin{aligned} dJ(\Omega)[\boldsymbol{V}] = &-\frac{1}{2}\int_\Sigma \left((\nabla u_D \cdot \mathbf{n})^2 + |\nabla u_R|^2\right) V_n \, ds \\ &- \int_\Sigma \beta \left(u_R(\boldsymbol{V} \cdot \nabla u_R) + |\nabla u_R|^2 \operatorname{div}_\Sigma \boldsymbol{V} - u_R q_R \operatorname{div}_\Sigma \boldsymbol{V}\right) ds \\ &+ \int_\Omega A\nabla u_R \cdot \nabla q_R \, dx + \int_\Sigma \lambda \left(\nabla u_R \cdot \boldsymbol{V} + u_R \operatorname{div}_\Sigma \boldsymbol{V} - q_R \operatorname{div}_\Sigma \boldsymbol{V}\right) ds \\ &:= I_1 + I_2 + I_3 + I_4. \end{aligned}$$

[12] A proof of this formula – also referred to as surface integration by parts formula – can be found in [49].

We apply formula (45) to rewrite the integrals I_2 and I_4 as follows:

$$\begin{aligned} I_2 &= \int_\Sigma (\partial_{\mathbf{n}} q_R (\nabla u_R \cdot \boldsymbol{V}) - \beta u_R (\nabla q_R \cdot \boldsymbol{V}))\, ds \\ &\quad + \int_\Sigma \beta (q_R \partial_{\mathbf{n}} u_R + u_R \partial_{\mathbf{n}} q_R + u_R q_R \operatorname{div}_\Sigma \mathbf{n})\, V_n \, ds \\ &\quad - \int_\Sigma \left(2\beta u_R \partial_{\mathbf{n}} u_R + \beta \operatorname{div}_\Sigma \mathbf{n} |\nabla u_R|^2\right) V_n \, ds \\ I_4 &= \int_\Sigma \lambda \left((\partial_{\mathbf{n}} u_R + u_R \operatorname{div}_\Sigma \mathbf{n} - \partial_{\mathbf{n}} q_R - q_R \operatorname{div}_\Sigma \mathbf{n})\, V_n + \nabla q_R \cdot \boldsymbol{V}\right) ds \end{aligned}$$

Meanwhile, since $q_R \in H^2(\Omega)$, then by (42), we can express I_3 in the following form

$$I_3 = - \int_\Sigma (\partial_{\mathbf{n}} q_R (\boldsymbol{V} \cdot \nabla u_R) + \partial_{\mathbf{n}} u_R (\boldsymbol{V} \cdot \nabla q_R) - (\nabla u_R \cdot \nabla q_R) V_n)\, ds.$$

Adding the above integrals to I_1, and then using the identities $|\nabla u_R|^2 = (\nabla u_R \cdot \mathbf{n})^2 + (\nabla u_R \cdot \tau)^2$, $\nabla u_R \cdot \nabla q_R = \nabla_\Sigma u_R \cdot \nabla_\Sigma q_R + \partial_{\mathbf{n}} u_R \partial_{\mathbf{n}} q_R$, and $\operatorname{div}_\Sigma \mathbf{n} = \kappa$, and applying twice the equation $\partial_{\mathbf{n}} u_R + \beta u_R = \lambda$ on Σ, we get, after some rearrangements, the desired characterization of the shape derivative of J with the shape gradient $\widetilde{\mathcal{G}_\beta}$ given by (37). This proves the proposition. □

Remark 6 In [33], Bacani obtained the expression

$$\frac{1}{2}\left((\partial_{\mathbf{n}} u_R)^2 - (\partial_{\mathbf{n}} u_D)^2 + 2\lambda\kappa u_R - (\partial_\tau u_R)^2\right) - \beta u_R (\partial_{\mathbf{n}} u_R + \kappa u_R) - \beta u_R u_R',$$

for dJ via the chain rule approach, where u_R' solves the system of equations (see [50])

$$\begin{cases} -\Delta u_R' = 0 & \text{in } \Omega, \\ u_R' = 0 & \text{on } \Gamma, \\ \partial_{\mathbf{n}} u_R' + \beta u_R' = \operatorname{div}_\Sigma (V_n \nabla_\Sigma u_R) + \lambda\kappa V_n - \beta (\partial_{\mathbf{n}} u_R + \kappa u_R)\, V_n & \text{on } \Sigma. \end{cases} \tag{46}$$

The (unique) existence of u_R' in $H^1(\Omega)$ is guaranteed for Ω of class $C^{2,1}$.

Similar to what has been pointed out in the proof of Proposition 2.2.5, we emphasize here that it is not practical to use u_R' in an iterative procedure to numerically solve (5) because (46) has to be solve for each field $\boldsymbol{V}$ which is apparently not easy. Nonetheless, there is actually an advantage in considering (46), particularly when the Robin coefficient β is chosen cleverly. In fact, if $\beta = \kappa$, then u_R' actually vanishes when Σ is the free boundary (cf. [50, Lem. 1]). In this way, those shape optimization reformulations of (1) that utilizes a state constraint with the Robin condition becomes easier to solve numerically via a second-order method. For some related works in this direction, see [20, 21, 50].

Remark 7 Observe that the computed expressions for dJ given by (9) and (37) differ in structure, particularly with respect to the adjoint system. In the minimax formulation, the corresponding adjoint system (10) to the min-sup problem (25), at $i = R$, is obtained by characterizing the saddle point of the Lagrangian functional G_R. Thus, it arises naturally in the proof of Proposition 2.1.1. On the other hand, the adjoint system introduced in the proof of Proposition 2.2.5 is constructed for the purpose of getting rid of $\widetilde{u_R}$ in (43). The appearance of the Eulerian derivative obviously could not be avoided – as opposed to applying the minimax formulation – due to the fact that we are considering a Robin condition on the free boundary. In spite of the difference in structure, we point out that both of the computed shape derivatives vanish at the shape solution Ω^* of (1), and thus satisfies the necessary optimality condition $dJ(\Omega^*)[\boldsymbol{V}] = 0$ for all $\boldsymbol{V}$. Meanwhile, regarding the use of the computed shape gradients in a numerical procedure, we shall see in subsection 3.2 that they actually provide nearly identical sequence of approximants to the optimal solution (with comparable speed of computation) of problem (5). Furthermore, we notice that we only introduced one adjoint system in the computation of the shape derivative of the cost via the Eulerian derivative of the states. In the minimax formulation, we obtained two adjoint systems, each correspond to the functional G_D and G_R. Even so, the adjoint variable p_D associated with G_D turns out to be zero on Σ, and therefore, does not appear in the expression for the shape gradient.

3 Numerical Approximation

As planned, we want to employ a gradient-based descent iterative procedure to numerically solve (5). The implementation is realized along the lines of the author's previous work, see [19–21], but with a few notable changes detailed in the following subsection (tailored for two dimensional cases).

To generate a sequence of approximants $\{\Omega_k\}_{k\geqslant 0}$ such that $J(\Omega_{k+1}) \leqslant J(\Omega_k)$, we will apply an explicit time linearization technique.

3.1 Numerical algorithm

Choice of descent direction. Let Ω_k be the shape of the domain at the kth iteration. At the $(k+1)$th iteration, the shape Ω_k is updated as $\Omega_{k+1} := \Omega_{t_{k+1}} = (id + t_k \boldsymbol{V}_k)\Omega_k$, for some small step size $t_k \geqslant 0$, where $\boldsymbol{V}_k$ represents the descent deformation field at the kth iterate.

For the descent direction, we can take $V_n = -\mathcal{G}_\beta$ on Σ. In practice, however, this choice may cause undesirable oscillations on the free boundary which is due, primarily, on the lack of sufficient regularity of $\mathcal{G}_\beta$ to preserve the smoothness of free boundary. To avoid such difficulty, we instead compute $\boldsymbol{V}$ as the solution in $Y(\Omega) := [V(\Omega)]^2$ of the variational problem

$$\mathbf{a}(\boldsymbol{V}, \boldsymbol{\varphi}) = -\int_\Sigma \mathcal{G}_\beta \mathbf{n} \cdot \boldsymbol{\varphi} \, ds, \quad \forall \boldsymbol{\varphi} \in Y(\Omega),$$

where $\mathbf{a}$ is a bounded coercive bilinear form on $Y(\Omega)$.[13] Here, we define

$$\mathbf{a}(\boldsymbol{\varphi}, \boldsymbol{\psi}) := \int_\Omega \nabla \boldsymbol{\varphi} : \nabla \boldsymbol{\psi} \, dx + c_\Sigma \int_\Sigma (\nabla_\Sigma \boldsymbol{\varphi} : \nabla_\Sigma \boldsymbol{\psi} + \boldsymbol{\varphi} \cdot \boldsymbol{\psi}) \, ds, \tag{47}$$

[13] Here, the notation $[\mathcal{X}(\cdot)]^2$ denotes the Sobolev space $[\mathcal{X}(\cdot)]^2 := \{\boldsymbol{\varphi} := (\varphi_1, \varphi_2) \mid \varphi_1, \varphi_2 \in \mathcal{X}(\cdot)\}$ and is equipped with the norm $\|\boldsymbol{\varphi}\|^2_{[\mathcal{X}(\cdot)]^2} = \|\varphi_1\|^2_{\mathcal{X}(\cdot)} + \|\varphi_2\|^2_{\mathcal{X}(\cdot)}$.

where $\boldsymbol{\varphi}, \boldsymbol{\psi} \in Y(\Omega)$, and $c_\Sigma \geqslant 0$ is some fixed real number. In this sense, the Sobolev gradient $\boldsymbol{V}$ (see, e.g., [51]) becomes a smoothed preconditioned extension of $-\mathcal{G}_\beta \mathbf{n}$ over the entire domain Ω.[14]

Computation of the discrete gradient. Now, taking into account the above computation of the extended-regularized descent direction, the sequence of discrete gradients $\{\boldsymbol{V}_k\}_{k \geqslant 0} := \{\boldsymbol{V}(\xi_k)\}_{k \geqslant 0}$ is generated according to the following elliptic PDE problem on Ω_k:

$$\begin{cases} \text{find } \boldsymbol{V}_k \in Y(\Omega_k) \text{ such that} \\ \qquad \mathbf{a}_k(\boldsymbol{V}_k, \boldsymbol{\varphi}) = -\displaystyle\int_{\Sigma_k} \mathcal{G}_{\beta k} \mathbf{n}_k \cdot \boldsymbol{\varphi} \, ds_k, \quad \forall \boldsymbol{\varphi} \in Y(\Omega_k), \end{cases} \tag{48}$$

where $\mathbf{a}_k$ is essentially the bilinear form $\mathbf{a}$ given in (47) with $\Omega = \Omega_k$.

As a result, the new computational domain Ω_{k+1} is obtained from the explicit update $\xi_{k+1} = \xi_k + t\boldsymbol{V}_k$. Note, however, that in general the resulting descent direction may not provide a decrease in the cost function. This suggests the necessity to employ a backtracking procedure for computing the step size t in the approximation process.

Remark 8 The case without the boundary integral in $\mathbf{a}$ already provides a good extension-regularization of $-\mathcal{G}_\beta \mathbf{n}$ in Ω. We only introduced the boundary integral expression with the weight parameter c_Σ to control and add more regularity to the resulting extension (higher values of c_Σ provides smoother extension of the gradient). On another note, we shall take $\beta = \kappa$ in all of our experiments we will conduct in subsection 3.2 focusing on a test case with strictly non-convex fixed boundary Γ. This choice of the Robin parameter might appear inappropriate when approximating an optimal shape solution that is strictly non-convex (and might actually violate the sign requirement for β during the approximation process), but, as we shall see in subsection 3.2, this seems to be not an issue (refer particularly to Figure 4(b)).

Remark 9 In our numerical experiments (particularly, in two dimensions), we will test our method with a similar problem setup considered in [53], and to this end, we will in fact compute the minimum curvatures of the computed optimal shape solutions for the given problems and compare them with the ones obtained in [53]. As we will be using FEMs, the curvatures are computed through the boundary nodes of the free boundaries whose degree of non-smoothness and irregularities, on the other hand, are of course influenced by the smoothing process that we have proposed above. It goes without saying that the discretization errors that accumulated throughout the iteration process affect the calculation of the minimum curvatures, and so, to minimize these expected errors, the introduction of the boundary integral in the definition of the bilinear form $\mathbf{a}$ with the smoothing parameter c_Σ is imperative in our case.

Step-size computation. The step size t_k can be decided in various ways. The simplest way is to set it to a fixed small positive real number. This, however,

[14] See [52] for more discussion about discrete gradient flows for shape optimization.

may result to a slow convergence behavior for the approximation process. A better way to update t_k is to apply a back-tracking procedure based on Armijo-Goldstein condition. Suppose that $0 \neq V_n = -\mathcal{G}_\beta \in L^2(\Sigma)$, then formally, for small $t > 0$, $J(\Omega_t) \simeq J(\Omega_0) + tdJ(\Omega_0)[\boldsymbol{V}] = J(\Omega_0) - t\|\mathcal{G}_\beta\|^2_{L^2(\Sigma_0)} < J(\Omega_0)$, where $\Omega_0 = \Omega$. Assuming that $J(\Omega_t) = (1-\mu)J(\Omega_0)$, for some $\mu \in (0,1)$, then t can be computed as $t = \mu J(\Omega_0)/\|\mathcal{G}_\beta\|^2_{L^2(\Sigma_0)}$. In our case, however, since $\boldsymbol{V}$ is defined via (48), the denominator has to be replaced by $\mathbf{a}_k(\boldsymbol{V}_k, \boldsymbol{V}_k)$ resulting to the formula $t_k = \mu J(\Omega_k)/\mathbf{a}_k(\boldsymbol{V}_k, \boldsymbol{V}_k)$ (see [21, p. 281]). That being so, it is natural to fix $\mu \in (0,1)$ at initialization and update the step size according to the following rule: take t_k as in the proposed formula whenever $J(\Omega_{k+1}) < J(\Omega_k)$; otherwise, scale down t_k and reinitialize the iteration with the previous shape Ω_k. We carry out the latter by scaling μ by half in the recalculation process. Essentially, at each iteration step k, t_k is set initially to $\mu J(\Omega_k)/\mathbf{a}_k(\boldsymbol{V}_k, \boldsymbol{V}_k)$. We also cut μ by half if reversed triangles are detected within the mesh update. In our experiments, we set $\mu = 0.5$.

The main steps in computing the kth domain Ω_k is given in the following algorithm:

Algorithm 1 Domain variation algorithm

```
1. Initialization Choose an initial shape $\Omega_0$.
2. Iteration For $k = 0, 1, 2, \ldots$,
    2.1 solve the state and adjoint state systems on $\Omega_k$;
    2.2 compute $\boldsymbol{V}_k$ via (48), and set $t_k = J(\Omega_k)/2\mathbf{a}_k(\boldsymbol{V}_k, \boldsymbol{V}_k)$;
    2.3 Update the current domain by $\Omega_{k+1} = (id + t_k\boldsymbol{V}_k)\Omega_k$.
3. Stop test Repeat the Iteration until convergence.
```

Stopping condition. To be more precise with the Stop test, we terminate the iterative procedure as soon as

$$\max\left(\sqrt{\mathbf{a}(\boldsymbol{V}, \boldsymbol{V})}, \|\boldsymbol{V}\|_{C(\Sigma)}\right) < \text{Tol}, \tag{49}$$

for some sufficiently small tolerance value Tol > 0, or after a finite number of iterations.

Evaluating the mean curvature. In Step 2.2 of the Iteration, the mean curvature κ of Σ needs to be computed at every iteration. Theoretically, for Ω of class C^2, there exists a unitary C^1 extension $\tilde{\mathbf{n}}$ of $\mathbf{n}$ such that the mean curvature may be defined as $\kappa = \nabla_\Sigma \cdot \mathbf{n} = \nabla \cdot \tilde{\mathbf{n}}$ (see [45, Prop. 5.4.8, p. 218] or [48, Lem. 16.1, p. 390]). Numerically, a way to compute the mean curvature, is to evaluate it as the divergence of some vector $\boldsymbol{N}$, where $\boldsymbol{N} \in [H^1(\Omega)]^2$ is an extension of $\mathbf{n}$[15] computed, for example, as the solution of the equation

$$c_{\mathrm{N}} \int_{\Omega} \nabla \boldsymbol{N} : \nabla \boldsymbol{\varphi} \, dx + \int_{\Sigma} (\nabla_{\Sigma} \boldsymbol{N} : \nabla_{\Sigma} \boldsymbol{\varphi} + \boldsymbol{N} \cdot \boldsymbol{\varphi}) \, ds = \int_{\Sigma} \mathbf{n} \cdot \boldsymbol{\varphi} \, ds, \tag{50}$$

for all $\boldsymbol{\varphi} \in [H^1(\Omega)]^2$, for some small real number $c_{\mathrm{N}} > 0$ (cf. [54]). Needless to say, this extension of the (outward) unit normal vector to Σ is not unitary, however, the approximation $\nabla \cdot \boldsymbol{N}$ of the mean curvature κ seems effective in our case. A more accurate numerical computation of the mean curvature could be given, but here we are satisfied with our results using this calculation. Besides, this approach can be extended without any difficulty in three dimensional setup and we do not actually care about the accuracy of this approximation of κ in the computation of the extension $\boldsymbol{V}$. So, in our numerical experiments, we will apply the said idea, and take $c_{\mathrm{N}} = 10^{-6}$. We remark here that the tangential gradient terms appearing in the boundary integrals above is added to create a more regular extensions than the ones used in previous works.

Remark 10 One of the challenges of existing mesh-based methods (e.g., FEMs) is the computationally expensive process of remeshing needed to carry out between optimization updates – especially in the case of three dimensions. In our numerical algorithm, however, we do not employ any sort of a remeshing procedure (as opposed to [19–21]), and as we shall see, our scheme even provides fast computation of the approximate solution though it is a domain discretization method. We point out, however, that in the case of two dimensions, remeshing is not costly. On a related subject concerning a fast numerical approach, but based on boundary element method which only requires discretization of the boundaries, see [53].

Remark 11 As remarked previously, we avoid the generation of a new triangulation of the domain at each iterative step. In practice, this is achieved by moving not only the boundary, but also the internal nodes of the mesh triangulation at every iteration. By doing so, the mesh only needs to be generated at initial iteration. To move the boundary and internal nodes simultaneously, we solve the discretized version of (48) and then move the domain in the direction of the resulting vector field scaled with the step size t_k. That is, we find $\boldsymbol{V}_k^h \in \mathbb{P}_1(\Omega_k^h)^d$ such that it solves the equation

$$\begin{cases} -\Delta \boldsymbol{V}_k^h = \mathbf{0} & \text{in } \Omega_k^h, \\ \boldsymbol{V}_k^h = \mathbf{0} & \text{on } \Gamma^h, \\ \nabla \boldsymbol{V}_k^h \cdot \mathbf{n}_k^h + c_{\Sigma} \boldsymbol{V}_k^h - c_{\Sigma} \Delta_{\Sigma} \boldsymbol{V}_k^h = -\mathcal{G}_{\beta}^k \mathbf{n}_k^h & \text{on } \Sigma_k^h. \end{cases}$$

where we suppose a polygonal domain $\overline{\Omega}_k^h$ and its triangulation $\mathcal{T}_h(\overline{\Omega}_k^h) = \{K_l^k\}_{l=1}^{N_e}$ (where K_l^k are the elements of the triangulation and N_e is the number of elements) are given, and $\mathbb{P}_1(\Omega_k^h)^d$ denotes the $\mathbb{R}^d$-valued piecewise linear function space on $\mathcal{T}_h(\overline{\Omega}_k^h)$. Then, we update the domain or equivalently, move the nodes of the mesh by defining Ω_{k+1}^h and $\mathcal{T}_h(\overline{\Omega}_{k+1}^h) = \{K_l^{k+1}\}_{l=1}^{N_e}$ respectively as $\overline{\Omega}_{k+1}^h := \left\{ \xi + t_k \boldsymbol{V}_k^h(\xi) \;\middle|\; \xi \in \overline{\Omega}_k^h \right\}$ and $K_l^{k+1} := \left\{ \xi + t_k \boldsymbol{V}_k^h(\xi) \;\middle|\; \xi \in K_l^k \right\}$, for all $k = 0, 1, \ldots$.

[15] Observe that this follows the same narrative in computing the Sobolev gradient $\boldsymbol{V}$ using (48).

Let us note that in Step 2.3 of Iteration, we have to find a suitable parametrization of Ω using a finite number of parameters. In accordance with the previous discussions, it is apparent that here we utilize the positioning of boundary (and internal) nodes of a partition into finite elements as design parameters or optimization variables.

Additional details of the computational setup. The numerical simulations we conduct below are obtained using the programming software FreeFem++ (see [55]). We use $\mathbb{P}_1$ finite element discretization to solve the variational problems. In deforming the shape of the domain, the function movemesh of FreeFem++ is utilized in the procedure. On a side note, a study on an adaptive FEM for shape optimization can be found in [56]. Lastly, all computations are performed on a 1.6 GHz Intel Core i5 Macintosh computer with 4GB RAM processors.

The computational setup stated above are tailored for the numerical example in two dimension which will be issued in the next subsection. For three dimensional problems, some details given in above discussion will be modified. The exact specifications will be given in subsection 3.4.

The computational or CPU times for the experiments will be shown in several tables below, and in all situations, the unit of time used to display the values are in 'seconds'.

3.2 Numerical examples in 2D

To assess the quality of the proposed scheme, we consider the same problem examined in [53] where $\Gamma = \partial D$ is defined as the boundary of a strictly non-convex domain D which is given by the following parametrization

$$D := \{(0.45 \cos\theta, 0.3 \sin\theta(1.25 + \cos 2\theta)), 0 \leqslant \theta \leqslant 2\pi\}.$$

In this experiment, we compute the optimal domain for all integers $\lambda = -1, -2, \ldots, -10$. We choose the circle $C(\mathbf{0}, 0.6)$ as the initial guess for all test cases where the interior and exterior boundaries are discretized with $N_{\text{int}} = 100$ and $N_{\text{ext}} = 120$ discretization points, respectively. Moreover, the initial computational mesh have the maximum and minimum mesh width $h_{\max} \approx 0.0623$ and $h_{\min} \approx 0.0155$ (see uppermost left plot in Figure 5). This choice of the initial mesh profile is course enough for the final mesh (of each test cases) not to have very flat triangles.

For $\lambda = -1, -2, -3, -4$, we stop the iteration process with Tol $= 0.001$ in (49), and the rest with Tol $= 0.01$. The computed exterior boundaries are shown in Figure 1 where the outermost boundary corresponds to $\lambda = -1$ and the innermost boundary to $\lambda = -10$. The evolution of the free boundary corresponding to the case when $\lambda = -3$ and $\lambda = -10$ are shown in Figure 2 with emphasis on the location of the boundary nodes at each iteration. Figure 3, on the other hand, depict the optimization histories for the same λ's as well as the histories of Sobolev gradient norms computed as $\|\boldsymbol{V}(\mathcal{G})\| := \max\{\sqrt{\mathbf{a}(\boldsymbol{V}, \boldsymbol{V})}, \|\boldsymbol{V}\|_{C(\Sigma)}\}$. Meanwhile, Figure 5 and Figure 6 respectively show that computational mesh profiles at selected iterates for the aforementioned cases.

We also computed the minimum curvature κ_2 of each of the computed optimal free boundary, the percentage error with respect to the computed minimum curvature κ_2^* in [53] (calculated as $|\kappa_2^* - \kappa_2|/|\kappa_2^*|$), and the over-all computing times (or CPU times) for each test cases. All these values which were obtained using the shape gradient $\mathcal{G}_\kappa$ given by (31) are tabulated in Table 1. Looking at the table, we see that our method provides accurate and fast computations of the optimal free boundary.

We also ran our algorithm with the shape gradient $\widetilde{\mathcal{G}}_\beta$ given by (37), and again with $\beta = \kappa$ (see (39)). For this experiment, however, we stopped the algorithm with Tol $= 0.001$ for $\lambda = -1, -2, -3$, and the rest with Tol $= 0.01$. The optimal free boundaries obtained for these experiments are identical with those plotted in Figure 1. The computational results, on the other hand, are summarized in Table 2. A superimposed comparison of the histories of curvatures for the cases $\lambda = -3$ and $\lambda = -10$ are plotted in Figure 4. In these plots, the solid lines with black-colored markers correspond to the results obtained using $\mathcal{G}_\kappa$ while the dotted lines with markers filled with white color correspond to the case when $\widetilde{\mathcal{G}}_\kappa$ is used in the algorithm. Notice that the computed curvatures at each iteration are nearly the same for the methods used. In fact, the evolutions of the free boundaries obtained via the shape gradient $\widetilde{\mathcal{G}}_\kappa$ are almost identical to those obtained using $\mathcal{G}_\kappa$ in the algorithm.

In terms of computational time per iteration, we can say that the two shape gradients provide comparable computational performance and are both effective in computing the optimal free boundary. Moreover, based from the experimental results, it seems that the optimization approach (2) is more robust compared to the classical Kohn-Vogelius method. The corresponding computational results obtained from implementing the latter formulation in our algorithm are provided in subsection 3.3 which contained further numerical experimentation focusing on the effect of the smoothing parameter c_Σ.

As mentioned in passing, a Newton scheme, which requires also the knowledge of the shape Hessian, was used in [53] in order to solve the same free boundary problem. The shape optimization formulation used in the said investigation, however, is different (i.e., it utilizes a Dirichlet-energy functional) and the numerical realization is carried out via boundary element methods. Although it cannot be compared directly to [53], we mention that the present algorithm requires less computational time per iteration (and overall computing time) to finish the optimization procedure than that of the aforementioned work. Of course, however, we cannot say that our method is faster than [53].

3.3 Parameter testing

To make our numerical findings in previous subsection more convincing, we tabulated here the numerical results obtained from using different values of c_Σ in our approximation scheme (under the same computational setup used in

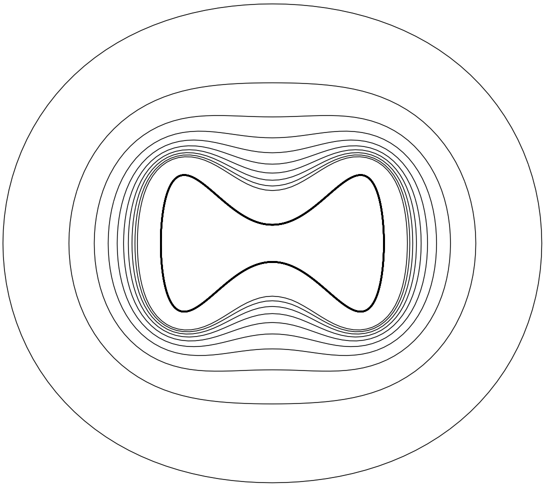


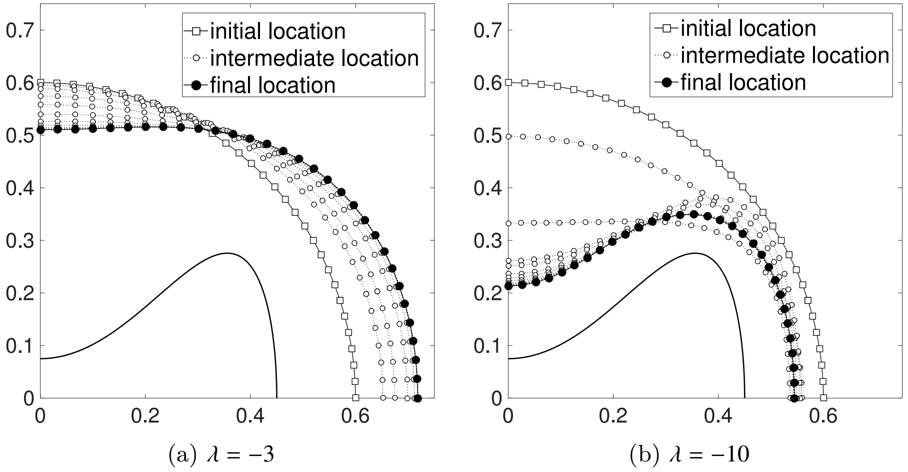
Figure 1: Optimal free boundaries Σ_*

$-\lambda$	c_Σ	κ_2	κ_2^* [53]	% error	iter.	CPU
1	4.100	0.683962	0.69	0.88	15	4.83
2	0.055	0.299734	0.30	0.09	14	4.11
3	0.160	-0.469733	-0.47	0.06	15	4.17
4	0.020	-1.399963	-1.40	0.00	12	3.83
5	0.050	-2.340385	-2.34	0.02	12	3.53
6	0.265	-3.196068	-3.19	0.19	13	3.80
7	0.355	-3.897400	-3.90	0.07	19	5.34
8	0.160	-4.591331	-4.59	0.03	18	5.11
9	0.060	-5.210978	-5.21	0.02	15	4.15
10	0.045	-5.497944	-5.52	0.40	19	5.94

Table 1: Computational results obtained using $\mathcal{G}_\kappa$ given by (31)

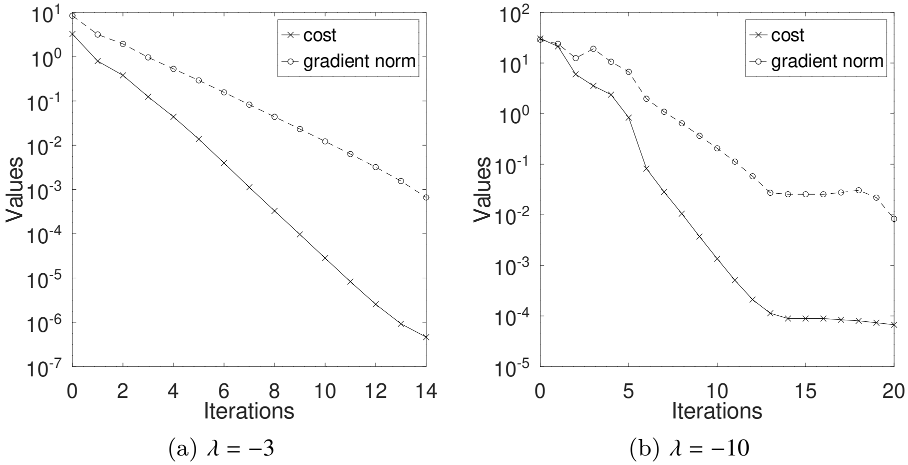
$-\lambda$	c_Σ	κ_2	κ_2^* [53]	% error	iter.	CPU
1	4.300	0.683757	0.69	0.90	15	4.86
2	0.045	0.300272	0.30	0.09	14	4.14
3	0.160	-0.469907	-0.47	0.02	15	4.14
4	0.025	-1.399385	-1.40	0.04	12	3.30
5	0.050	-2.340234	-2.34	0.01	12	3.42
6	0.225	-3.185848	-3.19	0.13	12	3.62
7	0.365	-3.895794	-3.90	0.11	19	5.20
8	0.175	-4.593019	-4.59	0.07	18	5.21
9	0.050	-5.169641	-5.21	0.77	17	5.74
10	0.045	-5.509279	-5.52	0.19	20	5.53

Table 2: Computational results obtained using $\widetilde{\mathcal{G}}_\kappa$ given by (39)



(a) $\lambda = -3$ (b) $\lambda = -10$

Figure 2: Evolution of the free boundary (with emphasis on boundary nodes) for the case when $\lambda = -3$ and $\lambda = -10$ plotted on the first quadrant

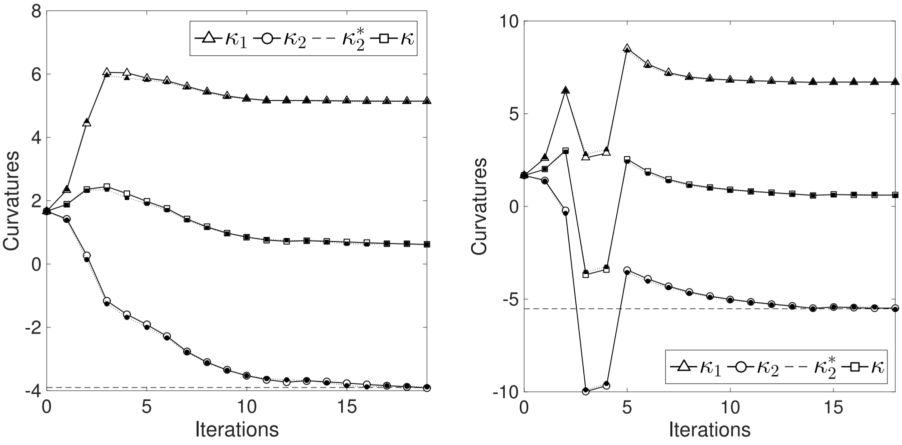


(a) $\lambda = -3$ (b) $\lambda = -10$

Figure 3: Histories of the cost values and norms of the Sobolev gradients corresponding to the case $\lambda = -3$ (left plot) and $\lambda = -10$ (right plot)

the previous subsection). The main point of the informations shown in tables Table 3–5 is to numerically support our claim that our proposed approach (2) of (1) provides a somewhat more robust method in approximating a solution to (1) compared to the classical Kohn-Vogelius method.

Here, we tested our algorithm with the shape gradients $\mathcal{G}_\kappa$, $\widetilde{\mathcal{G}}_\kappa$, and $\mathcal{G}_0 = \widetilde{\mathcal{G}}_0$ given by (31), (39), and (30), respectively. The choice of c_Σ that gives the most accurate minimum curvature (in comparison with the ones obtained in [53])



(a) $\lambda = -3$ (b) $\lambda = -10$

Figure 4: Histories of curvatures of the free boundaries plotted in Figure 2.

among the ones projected in the tables are highlighted in light gray color. For Table 3 and Table 4, the results are obtained with the same computational setup and assumptions given in the previous subsection. Based from these tables, it appears that both the shape gradients $\mathcal{G}_\kappa$ and $\widetilde{\mathcal{G}}_\kappa$ provide stable and accurate approximation of the optimal shape solution – in the sense that the approximate optimal domains computed for $\lambda = -1, \ldots, -10$ with $\Gamma := \partial D$ have the corresponding minimum curvatures almost equal to the ones computed in [53]. Meanwhile, the results summarized in Table 5 were obtained using the same tolerance values with the case when $\widetilde{\mathcal{G}}_\kappa$ is utilized in the algorithm (i.e., Tol = 0.001 is used for $\lambda = -1, -2, -3$, and Tol = 0.01 for the rest of the λ's considered).

In the experiments performed using $\mathcal{G}_0$, we noticed that our algorithm works well on dealing with the cases $\lambda = -4, \ldots, -10$. However, it seems that with $\mathcal{G}_0$, it is quite difficult to fine-tune the smoothing parameter c_Σ to be used in the algorithm and get an accurate approximation of the optimal domains having an error of less than 1% for the minimum curvature of the free boundary. This is contrary to the case of using either $\mathcal{G}_\kappa$ or $\widetilde{\mathcal{G}}_\kappa$ in the algorithm where we experience no issue in choosing the best value for c_Σ. Moreover, for the cases $\lambda = -1, -2$, and -3, we observed that it is difficult to achieve a good approximation of the optimal domain at the current computational setup, specifically with $\Sigma_0 = C(\mathbf{0}, 0.6)$ (see the highlighted result for $\lambda = -3$ which were obtained using the same computational setup for $\lambda = -4, \ldots, -10$). Apparently, the cost function J_0 becomes insensitive (with respect to large deformation) after several iterations for the cases $\lambda = -1, -2$, and -3. Figure 7 plots the histories of the cost values and Sobolev gradient norms – computed as $\|\boldsymbol{V}(\mathcal{G})\| := \max\{\sqrt{\mathbf{a}(\boldsymbol{V}, \boldsymbol{V})}, \|\boldsymbol{V}\|_{C(\Sigma)}\}$ – for the case $\lambda = -3$. As a result, the algorithm using $\mathcal{G}_0$ as the shape gradient converges prematurely and achieves a

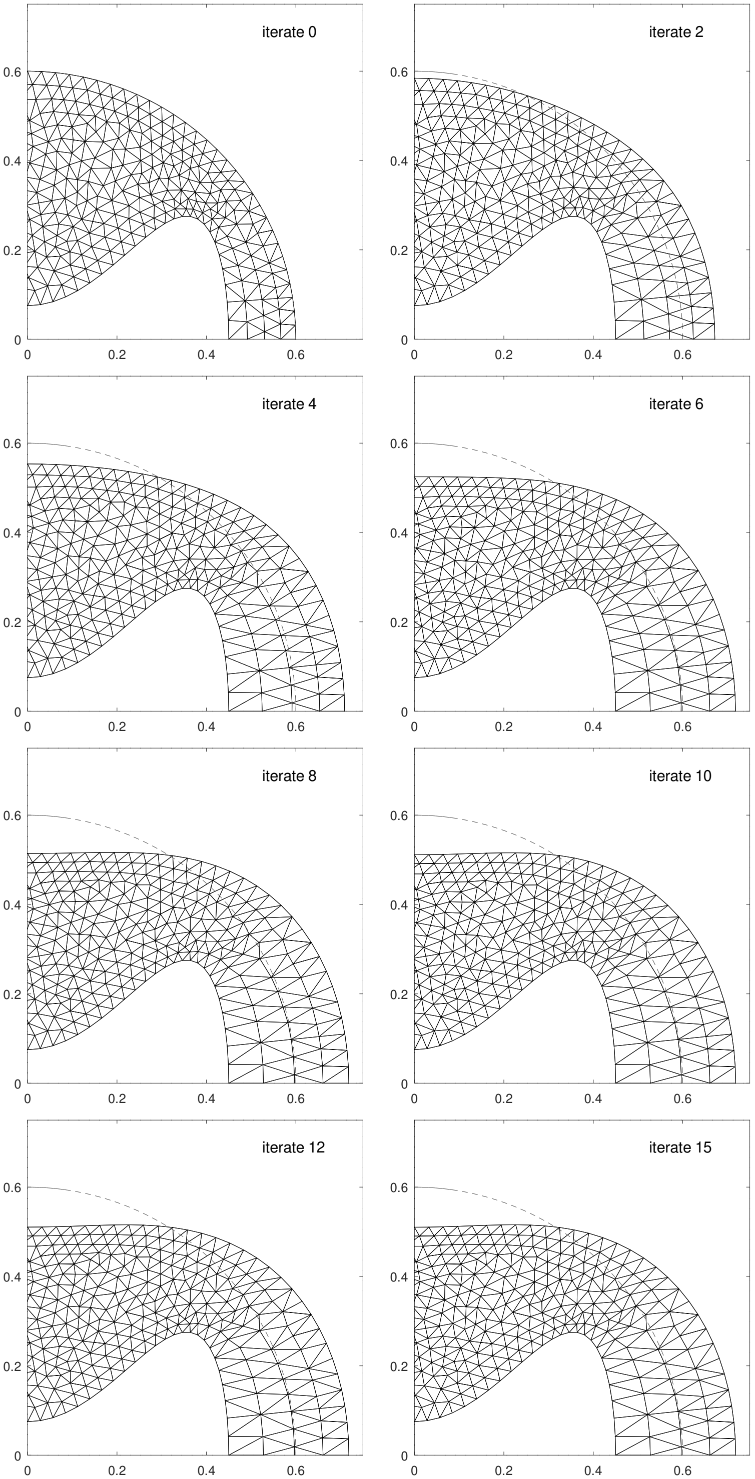


Figure 5: Mesh history at selected iterates when $\lambda = -3$

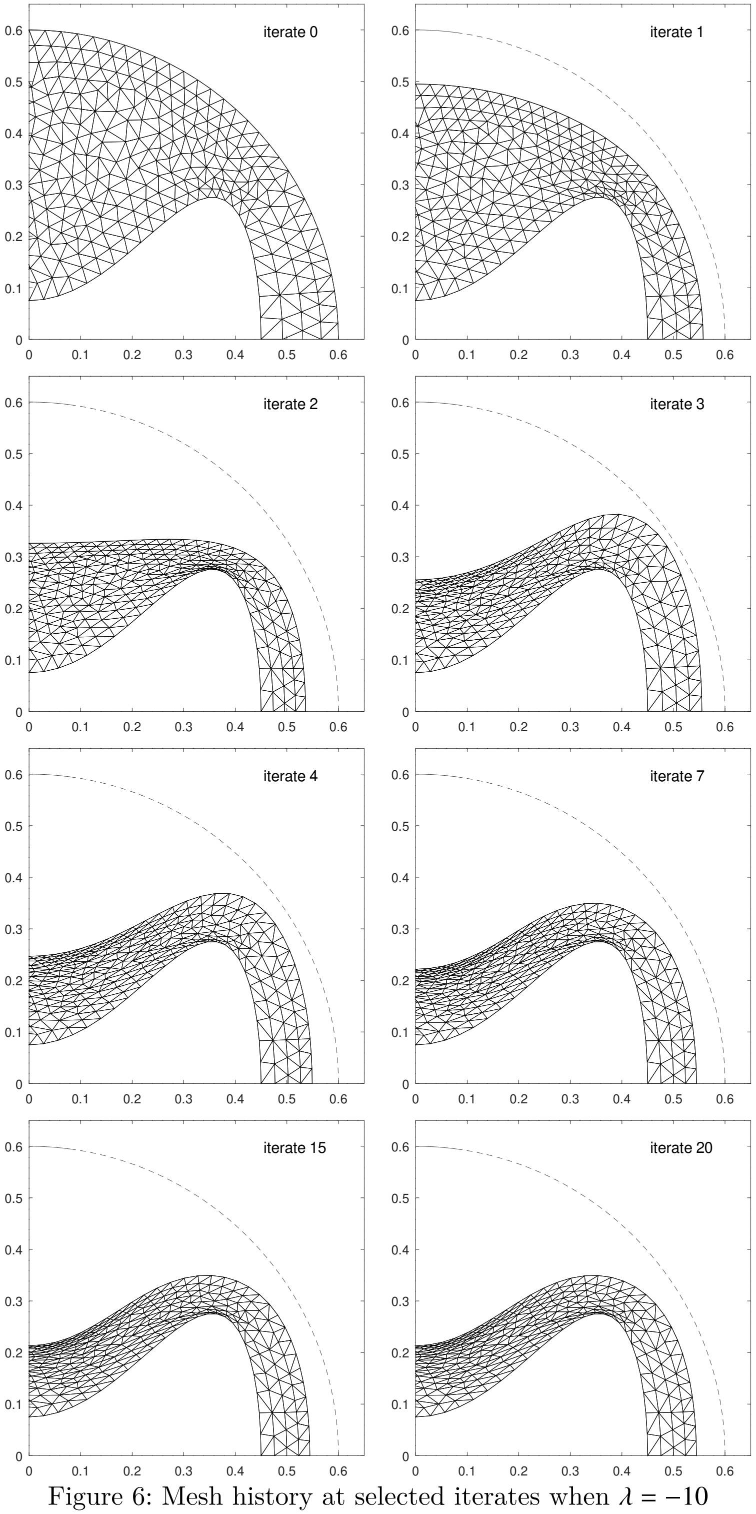
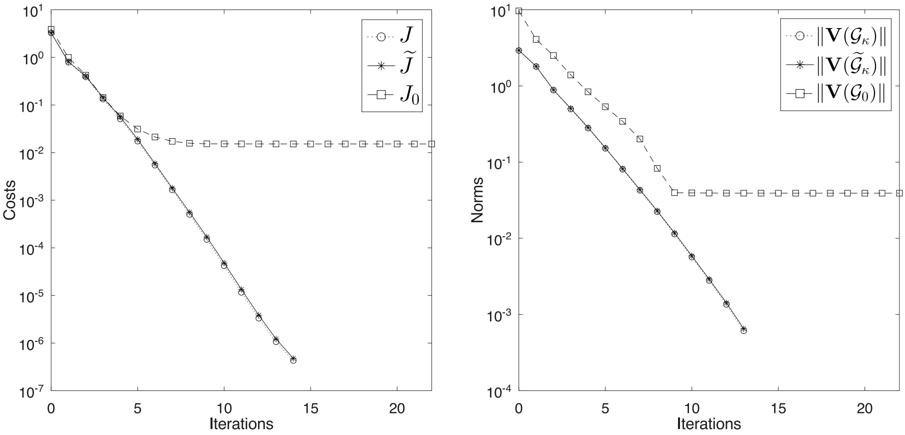


Figure 6: Mesh history at selected iterates when $\lambda = -10$

less accurate optimal domain for the said values of λ (see Figure 7(a)). Nevertheless, with the initial guess Σ_0 replaced by $C(\mathbf{0}, 1.0)$, we get an accurate approximation of the optimal free boundaries with respect to their corresponding minimum curvatures that we compared with those obtained in [53]. Based on this, it seems that the shape optimization approach (2) is somewhat more robust compared to the classical Kohn-Vogelius method when utilized in a Lagrangian-type gradient-based numerical scheme using finite element methods – such as ours. Even so, our numerical scheme, regardless of which shape gradient $\mathcal{G}_\kappa$, $\widetilde{\mathcal{G}_\kappa}$, or $\mathcal{G}_0 = \widetilde{\mathcal{G}_0}$ is used in the algorithm, provides fast computations of optimal free boundaries for the given problem, as evident in the computational times shown in tables 3–5.

Overall, we see that the proposed approach and the Kohn-Vogelius method nearly have the same computational times needed to complete the optimization procedure. However, in some cases, we see the advantage of employing the proposed formulation since it requires less CPU time for larger values of λ (i.e., cases when $\lambda = -3, -2, -1$).



(a) Histories of the cost values (b) Histories of Sobolev gradient norms

Figure 7: Histories of the cost values and norms of the Sobolev gradients corresponding to the case $\lambda = -3$ with $\Gamma_0 = C(\mathbf{0}, 0.6)$

3.4 Numerical examples in 3D

In this subsection, we test our algorithm in solving three dimensional cases. We first test the methods to a simple axisymmetric 3D-case with an analytical solution (of convex shape) and then to more complex cases (with a concavity and is only Lipschitz smooth) where an approximation procedure such as Algorithm 1 is better suited.

$-\lambda$	c_Σ	κ_2	κ_2^* [53]	% error	iter.	iter. [53]	CPU	CPU [53]
	3.900	0.683798		0.90	15		4.19	
	4.000	0.683703		0.91	15		4.74	
1	4.100	0.683962	0.69	0.88	15	9	4.83	92
	4.200	0.683598		0.93	15		4.78	
	4.300	0.683894		0.88	15		4.75	
	0.040	0.297898		0.70	14		4.13	
	0.045	0.298734		0.42	14		4.14	
	0.050	0.299600		0.13	14		4.19	
2	0.055	0.299734	0.30	0.09	14	7	4.11	73
	0.060	0.300663		0.22	14		4.15	
	0.150	-0.471218		0.26	15		4.10	
	0.155	-0.469105		0.19	15		4.16	
3	0.160	-0.469773	-0.47	0.06	15	7	4.17	79
	0.165	-0.470371		0.08	15		4.31	
	0.170	-0.471018		0.22	15		4.21	
	0.015	-1.407416		0.53	12		3.87	
	0.020	-1.399963		0.00	12		3.84	
4	0.025	-1.398331	-1.40	0.12	12	6	3.39	70
	0.030	-1.394629		0.38	12		3.34	
	0.035	-1.393288		0.48	12		3.30	
	0.045	-2.345335		0.23	12		3.45	
5	0.050	-2.340385	-2.34	0.02	12	7	3.53	89
	0.055	-2.338277		0.07	12		3.42	
	0.060	-2.336251		0.16	12		3.34	
	0.065	-2.334961		0.22	12		3.40	
	0.245	-3.202245		0.38	13		3.96	
6	0.250	-3.199881	-3.19	0.30	13	6	3.79	80
	0.255	-3.200115		0.32	13		3.82	
	0.260	-3.198089		0.25	13		3.79	
	0.265	-3.196068		0.19	13		3.80	
	0.340	-3.907415		0.19	18		5.09	
	0.345	-3.930059		0.77	22		6.42	100
	0.350	-3.909607		0.25	19		5.38	
7	0.355	-3.897400	-3.90	0.07	19	8	5.34	
	0.360	-3.916318		0.42	20		5.91	
	0.150	-4.617258		0.59	18		5.05	
	0.155	-4.568364		0.47	17		4.80	
8	0.160	-4.591331	-4.59	0.03	18	7	5.11	92
	0.165	-4.579966		0.22	19		5.68	
	0.170	-4.570499		0.42	16		4.64	
	0.045	-5.229026		0.37	15		4.11	
	0.050	-5.222427		0.24	15		4.09	
	0.055	-5.216895		0.13	15		4.10	
9	0.060	-5.210978	-5.21	0.02	15	7	4.15	90
	0.065	-5.205470		0.09	15		4.53	
	0.040	-5.555609		0.65	20		6.00	
10	0.045	-5.497944	-5.52	0.40	19	8	5.94	102
	0.050	-5.478265		0.76	18		6.03	
	0.055	-5.479253		0.73	18		5.28	
	0.060	-5.476906		0.78	16		5.02	

Table 3: Computational results obtained using the shape gradient $\mathcal{G}_\kappa$ given by (31) under different values of the smoothing parameter c_Σ

$-\lambda$	c_Σ	κ_2	κ_2^* [53]	% error	iter.	iter. [53]	CPU	CPU [53]
	3.900	0.683331		0.97	15		4.98	
	4.000	0.683590		0.93	15		4.87	
	4.100	0.683164		0.99	15		4.85	
	4.200	0.683461		0.95	15		4.94	
1	4.300	0.683757	0.69	0.90	15	9	4.86	92
	0.040	0.299439		0.19	14		4.25	
2	0.045	0.300272	0.30	0.09	14	7	4.14	
	0.050	0.300364		0.12	14		4.22	
	0.055	0.300501		0.17	14		4.20	73
	0.060	0.301423		0.47	14		4.23	
	0.150	-0.469596		0.09	15		4.31	
	0.155	-0.469284		0.15	15		4.16	
3	0.160	-0.469907	-0.47	0.02	15	7	4.14	79
	0.165	-0.470554		0.12	15		4.61	
	0.170	-0.469432		0.12	15		4.41	
	0.015	-1.408178		0.58	12		3.42	
	0.020	-1.404289		0.31	12		3.37	
4	0.025	-1.399385	-1.40	0.04	12	6	3.30	70
	0.030	-1.395728		0.31	12		3.35	
	0.035	-1.394467		0.40	12		3.68	
	0.045	-2.342605		0.11	12		3.42	
5	0.050	-2.340234	-2.34	0.01	12	7	3.42	89
	0.055	-2.336792		0.14	12		3.47	
	0.060	-2.333441		0.28	12		3.38	
	0.065	-2.331489		0.36	12		3.56	
	0.210	-3.201953		0.37	14		4.18	
	0.215	-3.204295		0.45	14		4.09	
	0.220	-3.200458		0.33	14		3.96	
6	0.225	-3.185848	-3.19	0.13	12	6	3.62	80
	0.230	-3.182508		0.23	12		3.67	
	0.355	-3.918150		0.47	19		5.48	
	0.360	-3.892460		0.19	19		5.49	
7	0.365	-3.895794	-3.90	0.11	19	8	5.20	100
	0.370	-3.882040		0.46	19		5.19	
	0.375	-3.884001		0.41	19		5.30	
	0.165	-4.584300		0.12	17		5.12	
	0.170	-4.599789		0.21	18		5.10	
8	0.175	-4.593019	-4.59	0.07	18	7	5.21	92
	0.180	-4.586269		0.08	16		4.52	
	0.185	-4.616170		0.57	22		6.33	
	0.060	-5.121169		1.71	18		4.00	
	0.065	-5.140184		1.34	17		5.39	
	0.070	-5.132413		1.49	16		4.87	
9	0.075	-5.169641	-5.21	0.77	17	7	4.83	90
	0.080	-5.121169		1.71	18		5.34	
	0.040	-5.505169		0.27	18		5.96	
10	0.045	-5.509279	-5.52	0.19	20	8	5.53	102
	0.050	-5.546782		0.49	18		5.07	
	0.055	-5.484489		0.64	21		6.19	
	0.060	-5.478805		0.75	21		6.03	

Table 4: Computational results obtained using the shape gradient $\widetilde{\mathcal{G}}_\kappa$ given by (39) under different values of the smoothing parameter c_Σ

$-\lambda$	c_Σ	κ_2	κ_2^* [53]	% error	iter.	iter. [53]	CPU	CPU [53]
	0.875	0.684941		0.73	24		18.73	
	0.900	0.685302		0.68	24		18.70	
	0.925	0.685664		0.63	24		19.43	
1	0.950	0.686037	0.69	0.57	25	9	19.00	92
	0.975	0.672496		2.54	11		16.86	
	0.095	0.286868		4.38	21		17.94	
	0.100	0.294453		1.85	20		17.69	
2	0.105	0.301428	0.30	0.47	20	7	18.37	73
	0.110	0.307203		2.40	20		18.15	
	0.115	0.311780		3.93	21		17.21	
	0.035	-0.457657		2.63	20		15.90	
	0.040	-0.454966		3.20	20		16.03	
3	0.045	-0.473090	-0.47	0.66	21	7	16.71	79
	0.050	-0.456539		2.86	21		16.04	
	0.055	-0.446174		5.07	21		16.80	
	0.000	-0.263695		43.89	19		7.53	
	0.025	-0.622141		32.37	26		8.64	
	0.050	-0.578362		23.06	23		7.77	
3	0.075	-0.576556	-0.47	22.67	23	7	7.86	79
	0.100	-0.590829		25.71	27		10.45	
	0.030	-1.436623		2.61	19		7.60	
4	0.035	-1.407973	-1.40	0.57	18	6	7.06	70
	0.040	-1.381807		1.30	19		7.24	
	0.045	-1.383943		1.15	18		7.35	
	0.050	-1.378626		1.53	20		7.36	
	0.030	-2.398938		2.52	18		6.65	
	0.035	-2.369408		1.26	18		6.90	
5	0.040	-2.341182	-2.34	0.05	17	7	6.40	89
	0.050	-2.293651		1.98	17		6.71	
	0.055	-2.271973		2.19	17		6.47	
	0.015	-3.262436		2.27	9		4.96	
6	0.020	-3.189927	-3.19	0.00	9		4.95	80
	0.025	-3.188130		0.06	16		5.72	
	0.030	-3.133338		1.78	16		5.74	
	0.035	-3.086156		3.26	15		5.65	
	0.040	-3.809256		2.33	9		5.41	
	0.045	-3.981180		2.08	9		5.83	
7	0.050	-3.922250	-3.90	0.57	8	8	5.29	100
	0.055	-3.869310		0.79	8		5.33	
	0.060	-3.659898		6.60	11		5.53	
	0.065	-4.301394		5.53	9		5.31	
	0.070	-4.294949		6.43	9		5.66	
8	0.075	-4.543171	-4.59	1.02	9	7	5.29	92
	0.080	-4.501519		1.92	9		5.21	
	0.085	-4.826300		5.15	10		5.14	
	0.035	-4.899750		5.95	9		4.96	
9	0.040	-5.163080	-5.21	0.90	9	7	5.08	90
	0.045	-5.107782		1.95	9		4.84	
	0.050	-5.058550		2.90	9		5.62	
	0.055	-5.016147		3.72	9		5.20	
	0.000	-5.601663		1.48	9		4.85	
	0.005	-5.444040		1.37	9		4.34	
	0.010	-5.613588		1.70	9		4.58	
10	0.015	-5.509235	-5.52	0.20	9	8	4.81	102
	0.020	-5.421308		1.79	9		4.31	

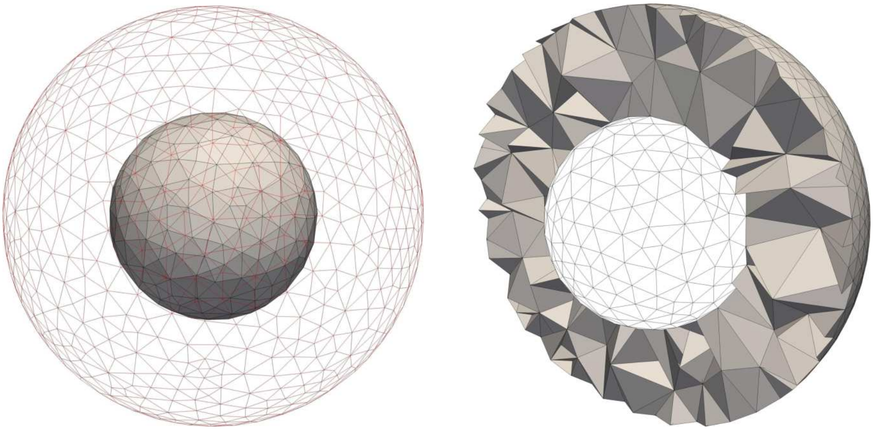
Table 5: Computational results obtained using the classical Kohn-Vogelius cost functional approach under different values of the smoothing parameter c_Σ

Because the performance of the shape gradients $\mathcal{G}_\beta$ and $\widetilde{\mathcal{G}}_\beta$ given in (9) and (37), respectively, are comparable as we have seen in previous subsection, we focus only on comparing the proposed method with the shape gradient $\widetilde{\mathcal{G}}_\beta$ and the classical Kohn-Vogelius method with the shape gradient $\mathcal{G}_0 = \widetilde{\mathcal{G}}_0$. In the experiments conducted below, we specifically compare the results obtain with the shape gradients $\mathcal{G}_0$, $\widetilde{\mathcal{G}}_\beta$, and $\widetilde{\mathcal{G}}_\kappa$, where β is a fixed (positive) real number.

To shorten many statements and for brevity, we refer to the proposed method with fixed β by DRb, and by DRk when $\beta = \kappa$. Meanwhile, we refer to the Kohn-Vogelius method simply by KV.

Example 1 (Axisymmetric case in 3D) Let us consider the spheres $\mathcal{S}(\mathbf{0}, r) := \{\xi \in \mathbb{R}^3 : |\xi| = r\}$ and $\mathcal{S}(\mathbf{0}, R) := \{\xi \in \mathbb{R}^3 : |\xi| = R\}$ centered at origin $\mathbf{0}$ with radius $r > 0$ and $R > r$, respectively. With $u(r) = 1$ and $u(R) = 0$, the solution to the Dirichlet problem (3) is exactly given by $u(\rho) = r(R - \rho)/[\rho(R - r)]$, $\rho \in (r, R)$ with normal derivative $\partial_\rho u(\rho) = -Rr/[\rho^2(R-r)]$. So, on the exterior surface, we have $\partial_\rho u(R) = -r/[R(R-r)] =: \lambda$. Thus, problem (1) with $\Gamma = \mathcal{S}(\mathbf{0}, r)$ and $\lambda = -r/[R(R - r)]$ has the unique exact free boundary solution $\Sigma_* = \mathcal{S}(\mathbf{0}, R^*)$.

For a concrete example, we let $r = 0.3$ and $R^* = 0.5$, giving us $\lambda = -3$, and choose the sphere $\mathcal{S}(\mathbf{0}, 0.6)$ for the initial guess. Moreover, we set $c_\Sigma = 0.1$ in the formula used to compute the descent direction, and take $\mu = 0.1$ (if not stated) for the calculation of the step size. The interior and exterior surface are discretized with maximum mesh size $h = 0.1$ and tetrahedrons of volume 0.01; see illustrated figure in Figure 8.



(a) Mesh profile (b) Cross section

Figure 8: Initial mesh profile for Example 1

Discussion for Example 1. The computational results for Experiment 1 are shown in Figure 9–Figure 11. Figure 9, in particular, shows the histories of

Hausdorff distances of the approximate shapes Σ_k with respect to the exact solution Σ_* for each of the three methods (with $\beta = 0.1$ for DRb). Based from the figures, it seems that with an appropriate choice of step size parameter μ, the convergence behavior of the methods being tested are comparable. More precisely, with $\mu = 2.0$ for DRb, $\mu = 1.5$ for DRk, and $\mu = 0.3$ for KV, we observed that the three approaches converge after (around) five iterations. In addition, we notice that KV is insensitive with large step sizes and it seems that the algorithm converges prematurely to a point that is far from the optimal solution. The CPU times (displayed in seconds) for the conducted experiments are shown in Table 6, and we see from this table that the CPU times needed by DRb and DRk to complete the iteration process is less compared to KV. This could be due to the part where the algorithm is performing the backtracking procedure. By this observation, we can say that – to some extent – the proposed method (2) by some means provides additional preconditioning of the descent direction.

Meanwhile, Figure 10 plots the histories of Hausdorff distances for DRb for some specific values of β taken on the interval $(0, 1]$ with $\mu = 0.1$. Looking at the plot, it appears that the convergence behavior of DRb is improved as β gets larger. So, since when $\beta = 0$ we recover KV from DRb, it appears that the modification of the Neumann problem of KV by the Robin problem in DRb somehow accelerates the convergence behavior of the classical method. The computational times for these experiments are tabulated in Table 7. In the table we notice that the average CPU times when $\beta > 0.5$ is slightly larger compared to when $\beta \in (0, 0.5]$. The slight difference in the average CPU times could again be due to the backtracking procedure employed in the algorithm.

Lastly, Figure 11 shows the mesh profiles of the computed shapes when $\mu = 2.0$ in DRb. Looking at the resulting figure, it appears that the approximation procedure is stable (in the sense that we do not observe any oscillations on the free surface during and after the approximation process) in the case when DRb is employed. We mention that we observe the same for the other two methods, particularly when $\mu < 1.8$ in DRk and $\mu < 0.5$ in KV.

Some modifications of the scheme. The computational setup used in the previous example is the same with the ones used in the previous section. For the rest of our examples, however, we simplify two parts of our algorithm. Firstly, in (48), we simply take $\mathbf{a}$ as the usual inner product in $[H^1(\Omega)]^2$-space. Secondly, in (50), we drop the tangential gradients and simply choose $c_N = 1$. Furthermore, we replace the formula $\mu J(\Omega_k)/\mathbf{a}(\boldsymbol{V}_k, \boldsymbol{V}_k)$ for computing the step size t_k with $\mu J(\Omega_k)/\|\boldsymbol{V}_k\|^2_{[L^2(\Sigma_k)]^2}$ to further increase the size of the step length. Such choice of the step size has already been used and found to be effective in [19].

We now proceed with our next test case which is slightly similar to the example problem considered in [38] where a Newton method for a Bernoulli's free boundary problem in three dimensions was proposed (see also the model problem [7]).

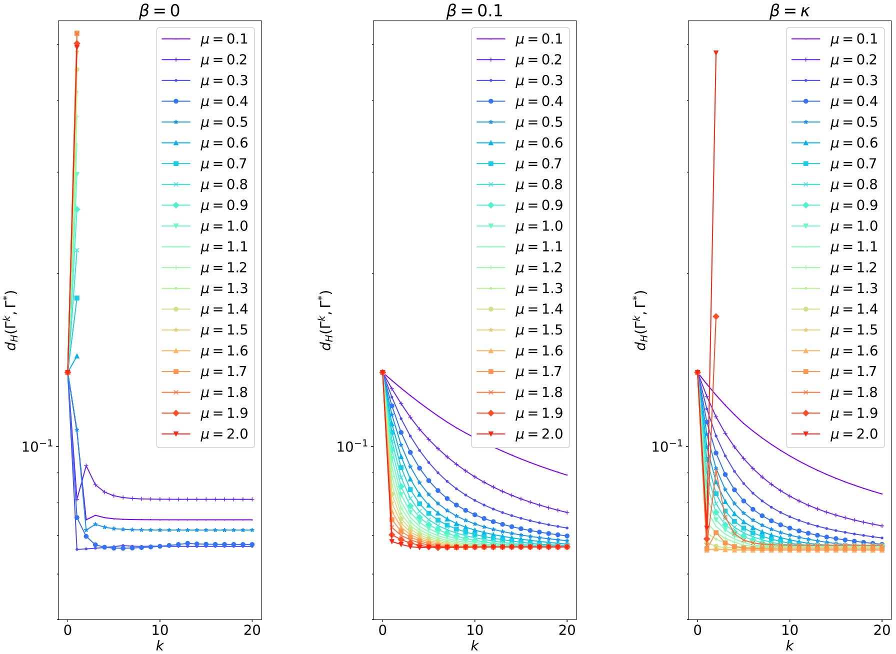


Figure 9: Histories of Hausdorff distances

μ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
DRb	19.68	18.66	18.75	18.80	18.57	18.95	19.22	18.70	21.06	20.17	21.65	22.58	23.47	23.45	23.15	24.90	24.27	24.00	30.73	25.69
DRk	18.92	18.37	18.49	18.27	19.00	19.51	22.31	24.81	24.73	23.13	26.16	26.17	25.89	28.26	28.21	28.10	27.62	28.24	-	-
KV	32.37	33.24	31.16	27.70	33.07	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 6: CPU times corresponding to the plots shown in Figure 9

β	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
CPU	19.95	20.26	19.31	19.88	19.65	20.24	20.53	21.28	21.11	21.60	23.82	25.60	25.57	23.97	24.76	23.30	24.20	24.80	25.39	26.84

Table 7: CPU times corresponding to the plots shown in Figure 10

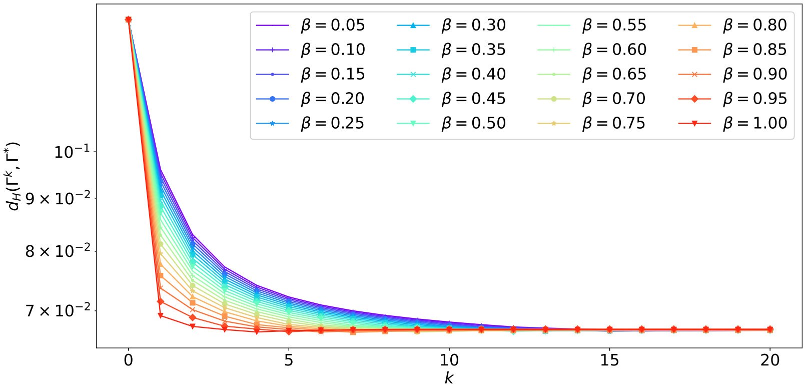
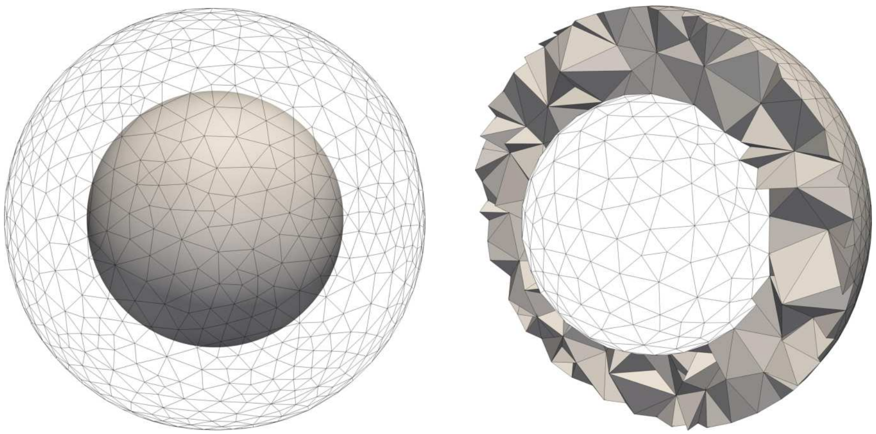


Figure 10: Histories of Hausdorff distances for $\beta = 0.05, 0.10, 0.15, \ldots, 1.00$.



(a) Mesh profile ($\beta = 0.1$, $\mu = 2.0$)

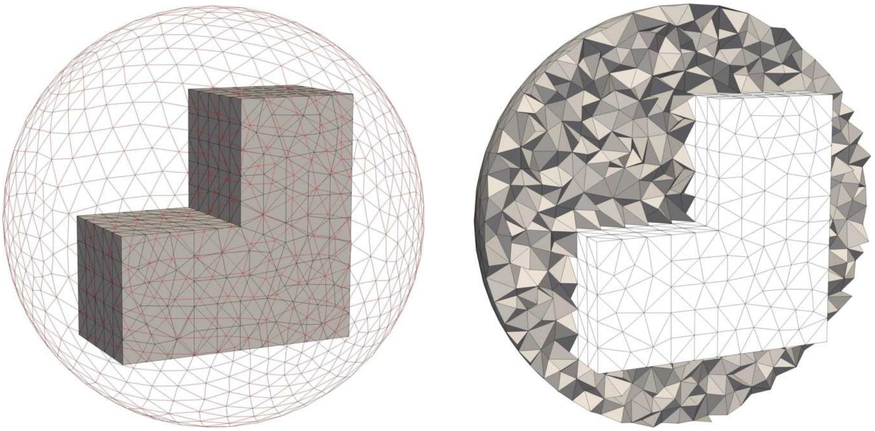
(b) Exterior surface ($\beta = 0.1$, $\mu = 2.0$)

Figure 11: Computed shape for Example 1 via DRb method

Example 2 (L-block figure) We consider Σ as the surface of an L-block figure, and consider the values $\lambda = -3, -2, -1$. The L-block domain is given as follows:

$$S = ([-0.5, 0.5] \times [-1, 1]^2) \setminus ([-0.5, 0.5] \times [0, 1]^2).$$

For all cases, we choose the sphere of radius 1.7 for the initial guess and (initially) discretize the computational domain with maximum mesh size $h_{\max} = 0.2$ for $\lambda = -3$ and $h_{\max} = 0.4$ for $\lambda = -2, -1$. In all cases, the (maximum) volume of the tetrahedrons are set to 0.001. See Figure 12 for an illustration of the initial mesh profile and a cross-sectional view of the computational mesh when $\lambda = -1$. Meanwhile, the step size parameter $\mu > 0$ is chosen small enough such that the algorithm is stable (does not converge prematurely, at least for DRb and DRk) and that there is no oscillations on the free boundary. We compare the approximation results obtained using DRb, DRk, and KV. For the DRb method, we fix β to the unit value (i.e. we take $\beta = 1$). We take the sphere $\mathcal{S}(\mathbf{0}, 1.7)$ as an initial guess for the experiments.



(a) Mesh profile

(b) Cross section

Figure 12: Initial mesh profile for Example 2 (in the case of $\lambda = -1$)

Discussion for Example 2. The results of the experiments are shown in the rest of the figures below.

Figure 13 shows the cross sectional views of the computed optimal free boundaries obtained via the DRb method where the innermost exterior boundary corresponds to $\lambda = -3$ and the outermost exterior boundary to $\lambda = -1$. The computational mesh corresponding to the case $\lambda = -3$ is also shown in the said figure.

Meanwhile, Figure 14, Figure 15, and Figure 16 respectively depict the mesh profile of the computed optimal exterior free boundaries for each $\lambda = -3, -2, -1$. The first, second, and third rows of plots in Figure 14–Figure 16 respectively show that computed exterior free boundaries obtained via DRb, DRk, and KV. Notice from these figures that the computed shapes for DRb and DRk almost coincide (as expected). However, the computed shape for KV seems more convex compared to the ones obtained through DRb and DRk. In fact, it is evident in Figure 17 which shows a superimposed comparison of DRb and KV (at different cross sections), that the concave parts of the computed optimal exterior surface under KV method is less pronounced than the case of DRb and DRk. Thus, it appears that under the current computational setup (with $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.7)$), the KV method provides a bit less accurate optimal free boundary when $\lambda = -1$. This scenario is somewhat similar to the case we experience in the experiments we performed in subsection 3.3 where we observe some difficulty in obtaining an accurate approximation of the optimal free boundary for the case $\lambda = -3$ when using the same classical method (refer to Table 5). Nevertheless, after repeating the approximation procedure, but now with $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$, we got a good agreement between DRb, DRk, and KV, see Figure 18 (see also Figure 19 for a superimposed comparisons of the cross sections of the approximate optimal shapes obtained using the proposed method DRb and the classical method KV).

The optimization histories as well as the histories of $[L^2(\Sigma)]^2$ norms of the Sobolev gradients are depicted in Figure 20 for the case when $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.7)$ and Figure 21 when $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$. Notice that in the latter figure, we observe a bit faster convergence behavior for the DRb and DRk methods than KV. Moreover, looking at Table 8, it appears that the computational time needed to complete the iteration process when employing DRb or DRk is generally less compared to KV.

To end our discussion, we summarize below our key observations in the above performed experiments:

- our proposed method DR is more sensitive than KV in terms of large variations;
- in some instances – depending on the choice of the initial guess – the KV method tends to converge prematurely to a less accurate optimal solution;
- the convergence speed of DR and KV can be made comparable by choosing appropriate value for the step size parameter μ;
- however, in general, KV seems to converge in a fewer number of steps than DR;
- nonetheless, as in the case of 2D, DR requires less computational time to finish the iteration procedure.

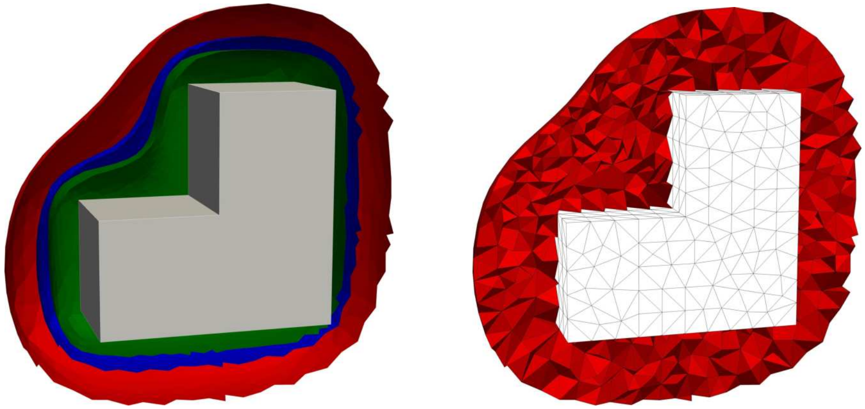
The latter two observations could be due to the effect of the backtracking procedure implemented in the algorithm which, on the other hand, is dependent on the computed maximal step size provided by each method at each iterations.

Overall, we found some merits in considering the proposed shape optimization formulation (2) of the exterior free boundary problem (1).

Lastly – although it cannot be compared directly – we remark that the computational cost of our proposed algorithm is less compared to the one used in [38]. Even so, this does not suggests that our proposed approach is absolutely faster than the method used in the aforementioned study.

λ	-3	-2	-1	-1
r	1.7	1.7	1.7	1.8
DRb	405	471	450	559
DRk	471	416	475	560
KV	312	477	482	605

Table 8: CPU times corresponding to the plots shown in Figure 14–Figure 16 (the cases when $\Sigma_0 = \mathcal{S}(\mathbf{0}, r) = \mathcal{S}(\mathbf{0}, 1.7)$), and Figure 18 ($\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$)



(a) Cross sectional view (for all λ's)

(b) Cross sectional view ($\lambda = -1$)

Figure 13: Cross sectional views of the computed shapes for the DRb method (innermost exterior boundary: $\lambda = -3$, outermost exterior boundary: $\lambda = -1$)

4 Concluding Remark

We proposed an efficient shape optimization scheme for the numerical resolution of the exterior Bernoulli problem. The formulation examined in this study can be viewed as a small generalization of the so-called Kohn-Vogelius method. Moreover, the Robin term introduced in the formulation somehow acts as an acceleration of the classical method which also provides further preconditioning of the descent direction. Numerical experiments showed that the proposed approach is fast and effective in solving the exterior Bernoulli free boundary problem – at least for the problems considered in this investigation.

Acknowledgments. The author acknowledges the support from JST CREST Grant Number JPMJCR2014. He also wishes to thank the two anonymous referees for their stimulating comments, constructive criticisms, and a number of useful suggestions which greatly improved the presentation of the paper.

Declarations

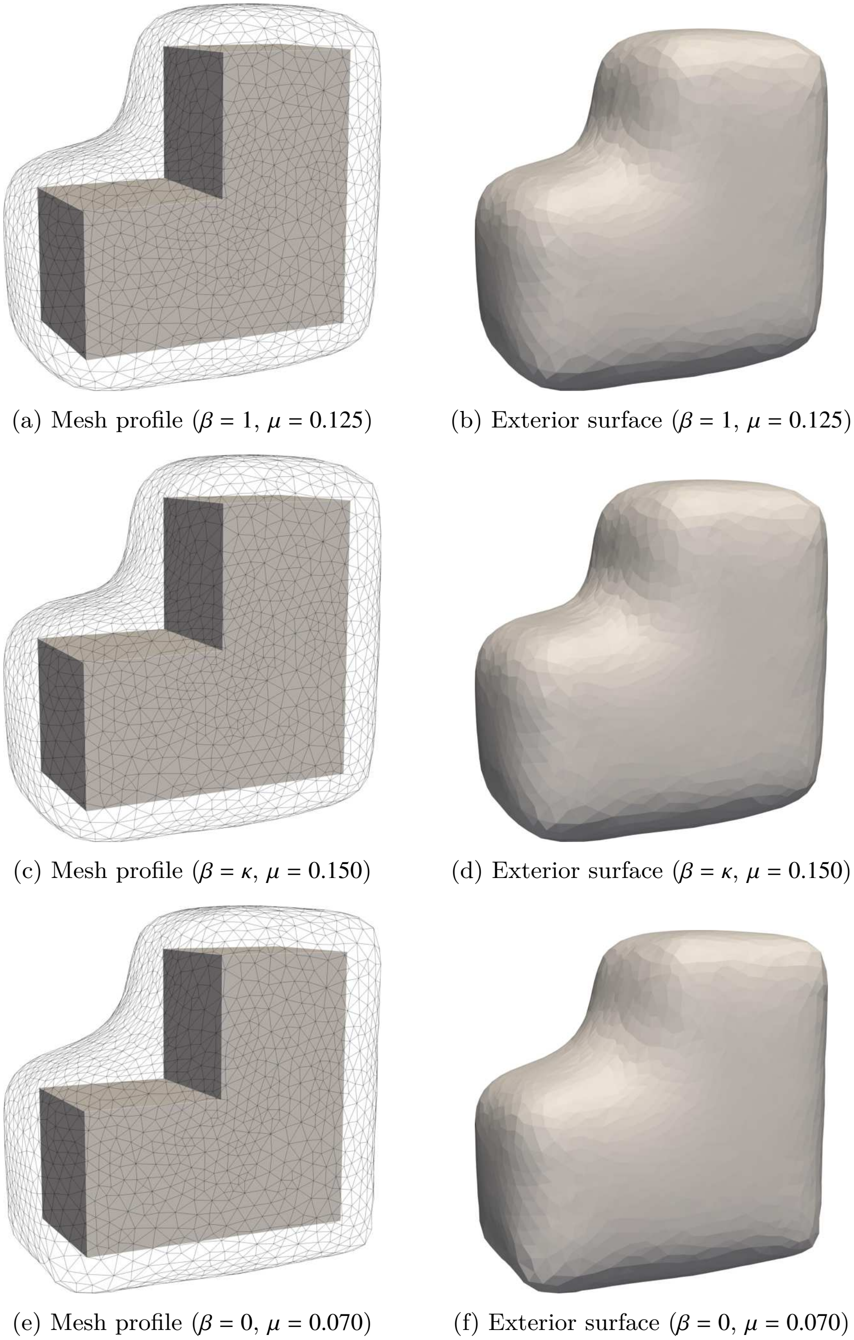
Data sharing. Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Conflict of interest. The author declare that there is no conflict of interest.

A Correa-Seeger Theorem

Let $\varepsilon > 0$ be a fixed real number and consider a functional

$$G : [0, \varepsilon] \times X \times Y \to \mathbb{R},$$



(a) Mesh profile ($\beta = 1$, $\mu = 0.125$)

(b) Exterior surface ($\beta = 1$, $\mu = 0.125$)

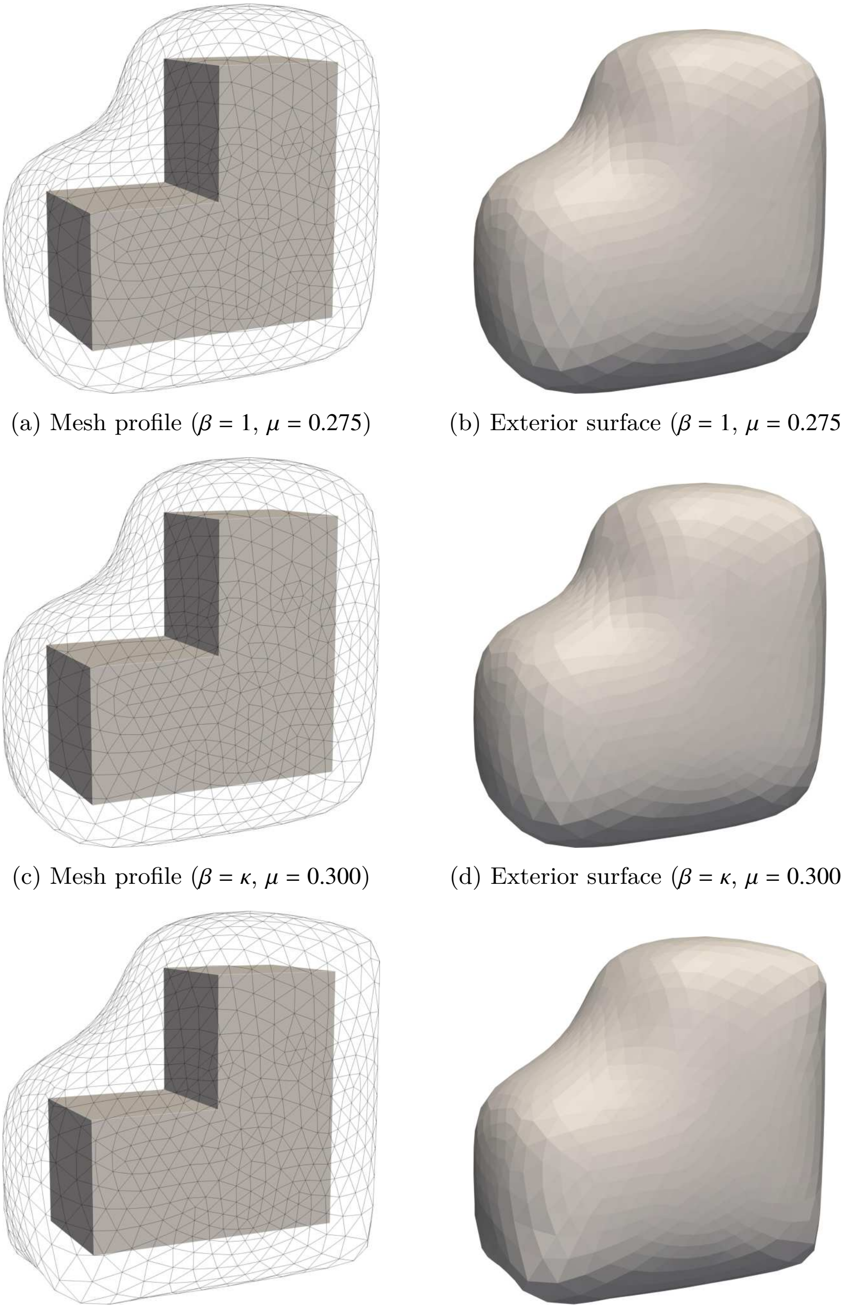
(c) Mesh profile ($\beta = \kappa$, $\mu = 0.150$)

(d) Exterior surface ($\beta = \kappa$, $\mu = 0.150$)

(e) Mesh profile ($\beta = 0$, $\mu = 0.070$)

(f) Exterior surface ($\beta = 0$, $\mu = 0.070$)

Figure 14: Computed shapes for Example 2 when $\lambda = -3$

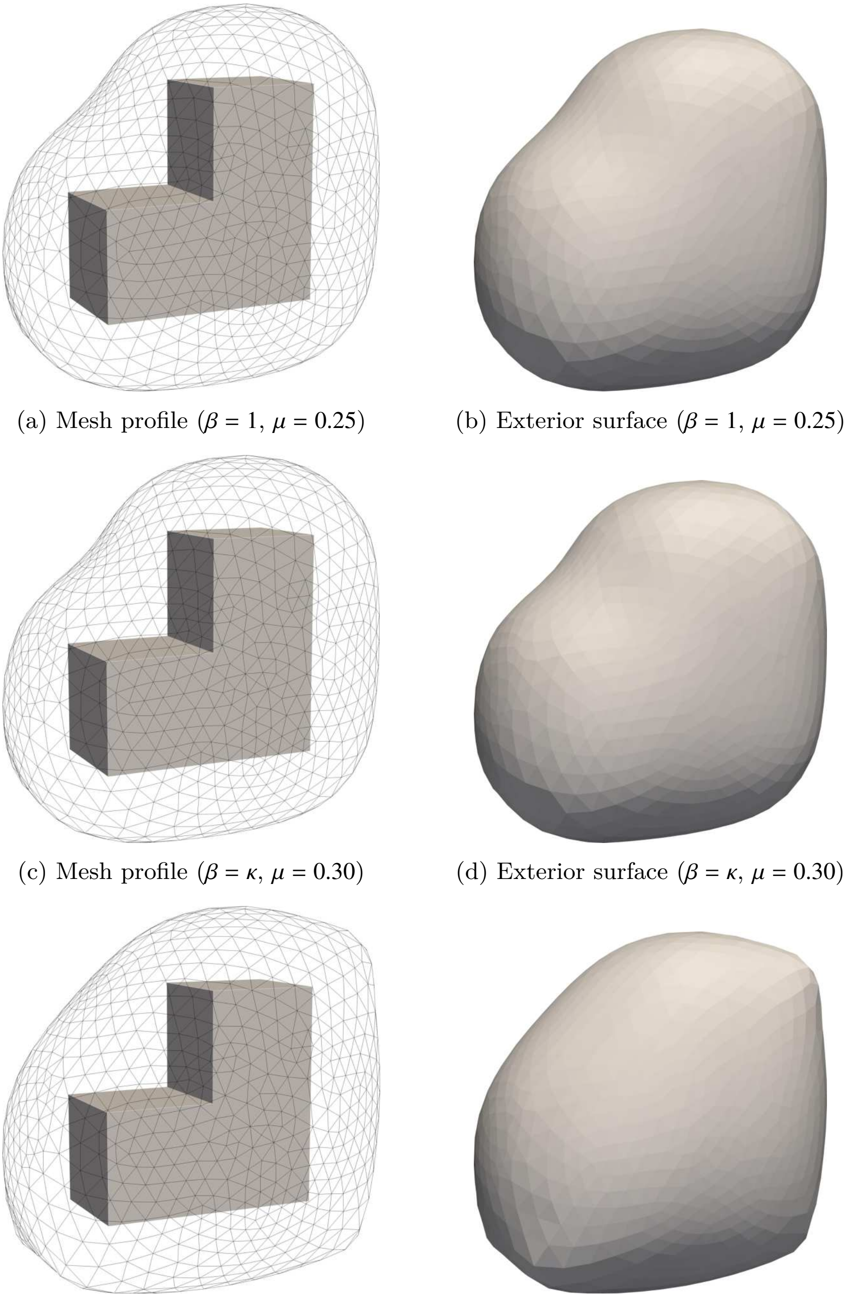


(a) Mesh profile ($\beta = 1$, $\mu = 0.275$) (b) Exterior surface ($\beta = 1$, $\mu = 0.275$)

(c) Mesh profile ($\beta = \kappa$, $\mu = 0.300$) (d) Exterior surface ($\beta = \kappa$, $\mu = 0.300$)

(e) Mesh profile ($\beta = 0$, $\mu = 0.175$) (f) Exterior surface ($\beta = 0$, $\mu = 0.175$)

Figure 15: Computed shapes for Example 2 when $\lambda = -2$



(a) Mesh profile ($\beta = 1$, $\mu = 0.25$)

(b) Exterior surface ($\beta = 1$, $\mu = 0.25$)

(c) Mesh profile ($\beta = \kappa$, $\mu = 0.30$)

(d) Exterior surface ($\beta = \kappa$, $\mu = 0.30$)

(e) Mesh profile ($\beta = 0$, $\mu = 0.20$)

(f) Exterior surface ($\beta = 0$, $\mu = 0.20$)

Figure 16: Computed shapes for Example 2 when $\lambda = -1$

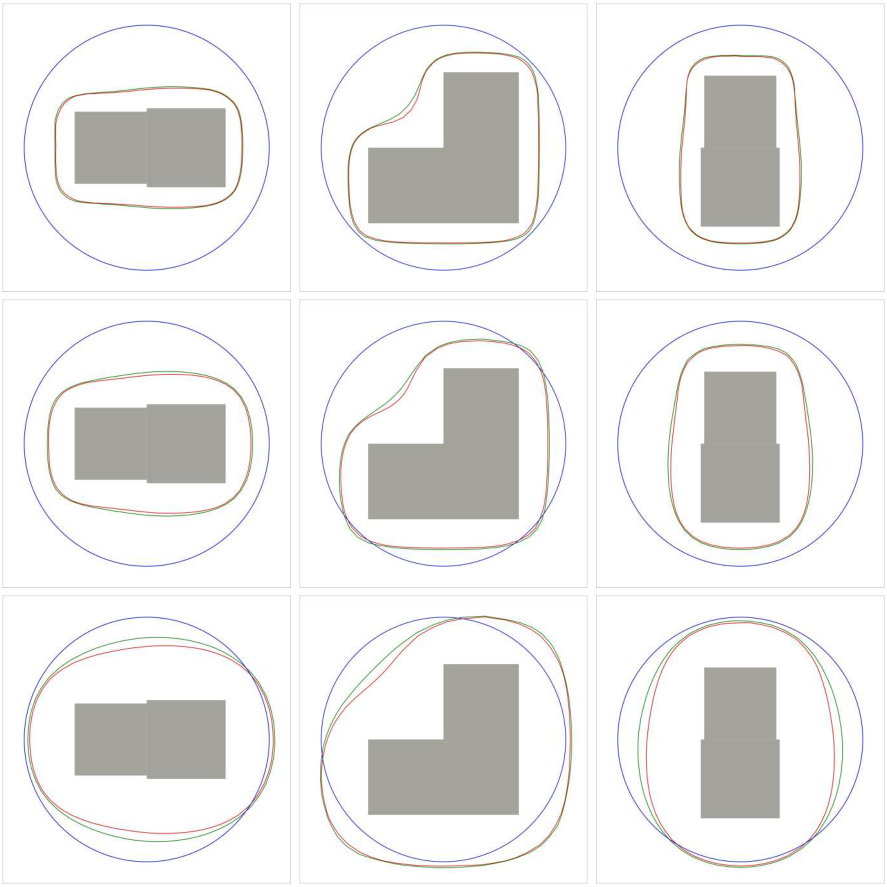


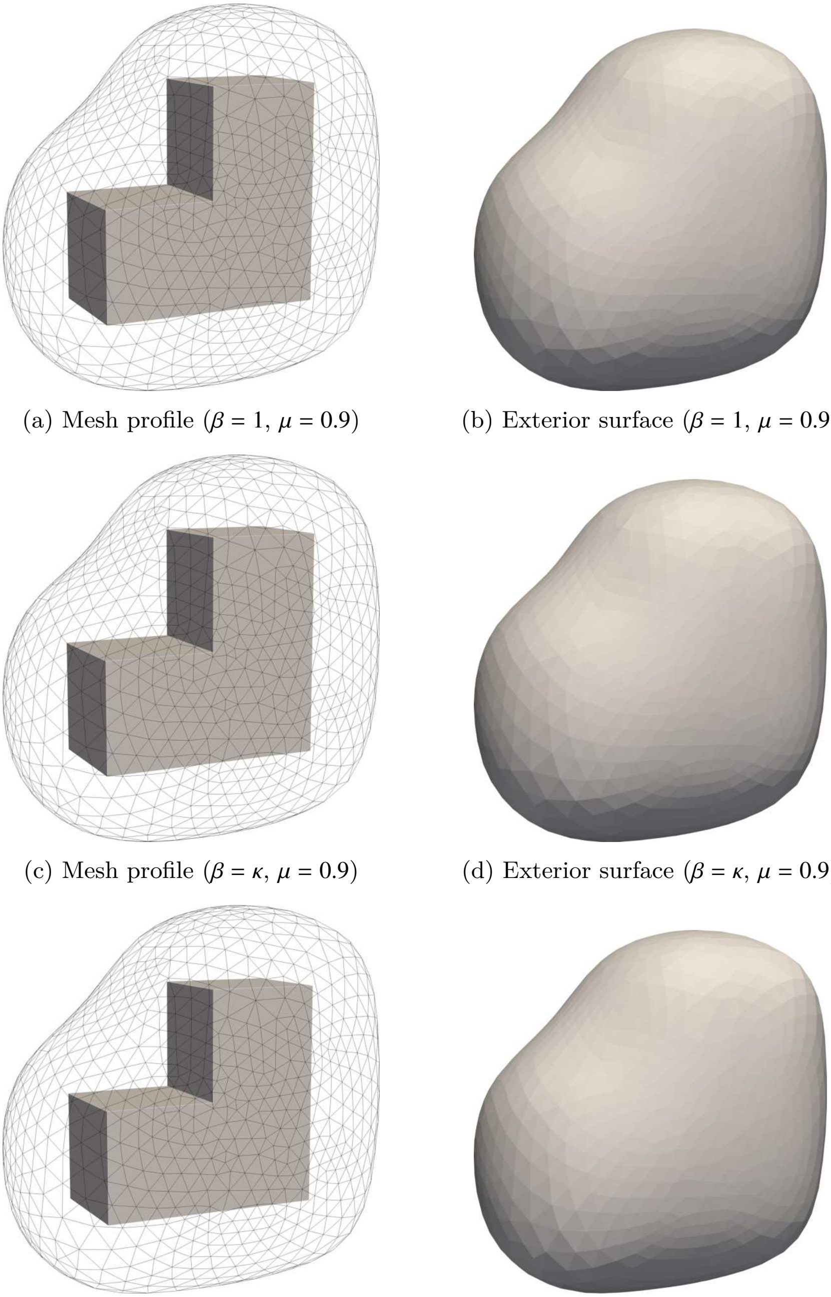
Figure 17: Cross comparisons of DRb (red lines) and KV (green lines) for $\lambda = -3$ (top row), -2 (mid row), -1 (bottom row) with $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.7)$ (blue lines)

for some topological spaces X and Y. For each $t \in [0, \varepsilon]$, we define

$$M(t) := \min_{x \in X} \sup_{y \in Y} G(t, x, y) \quad \text{and} \quad m(t) = \sup_{y \in Y} \min_{x \in X} G(t, x, y),$$

and the associated sets

$$X(t) := \left\{ \hat{x} \in X : \sup_{y \in Y} G(t, \hat{x}, y) = M(t) \right\}$$
$$Y(t) := \left\{ \hat{y} \in Y : \min_{x \in X} G(t, x, \hat{y}) = m(t) \right\}.$$



(a) Mesh profile ($\beta = 1$, $\mu = 0.9$)

(b) Exterior surface ($\beta = 1$, $\mu = 0.9$)

(c) Mesh profile ($\beta = \kappa$, $\mu = 0.9$)

(d) Exterior surface ($\beta = \kappa$, $\mu = 0.9$)

(e) Mesh profile ($\beta = 0$, $\mu = 0.3$)

(f) Exterior surface ($\beta = 0$, $\mu = 0.3$)

Figure 18: Computed shapes for Example 2 when $\lambda = -1$ with $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$

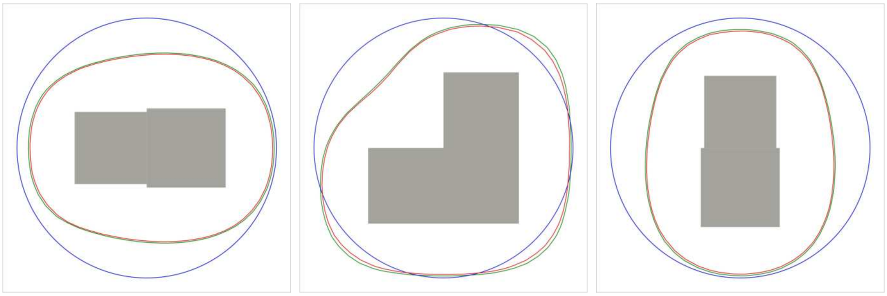
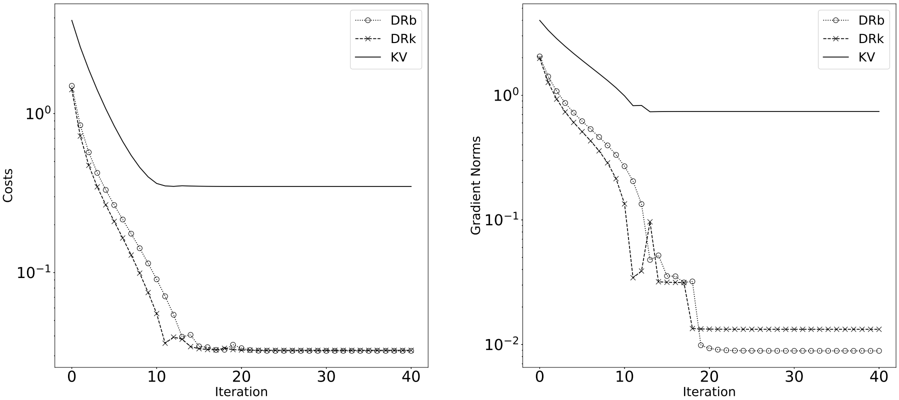


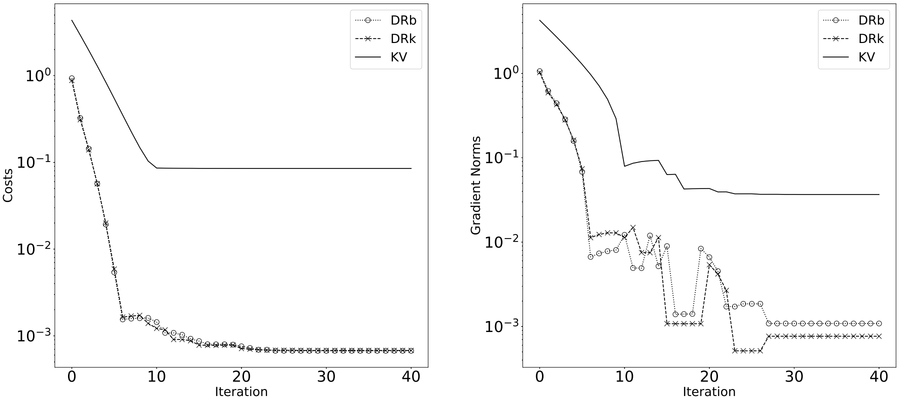
Figure 19: Cross comparison of DRb (red lines) and KV (green lines) for $\lambda = -1$ with $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$ (blue lines)



(a) Histories of the cost values

(b) Histories of Sobolev gradient norms

Figure 20: Computational results corresponding to $\lambda = -3$ when $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.7)$



(a) Histories of the cost values

(b) Histories of Sobolev gradient norms

Figure 21: Computational results corresponding to $\lambda = -3$ when $\Sigma_0 = \mathcal{S}(\mathbf{0}, 1.8)$

We introduce the set of saddle points

$$S(t) = \{(\hat{x}, \hat{y}) \in X \times Y : M(t) = G(t, \hat{x}, \hat{y}) = m(t)\},$$

which may be empty. In general, we always have the inequality $m(t) \leqslant M(t)$, and when $m(t) = M(t)$, the set $S(t)$ is exactly $X(t) \times Y(t)$.

We have the following theorem (see [35, Thm. 5.1, pp. 556–559]).

Theorem A.1 (Correa and Seeger, [43]) Let the sets X and Y, the real number $\varepsilon > 0$, and the functional $G : [0, \varepsilon] \times X \times Y \to \mathbb{R}$ be given. Assume that the following assumptions hold:

(H1) for $0 \leqslant t \leqslant \varepsilon$, the set $S(t)$ is non-empty;

(H2) the partial derivative $\partial_t G(t, x, y)$ exists everywhere in $[0, \varepsilon]$, for all $(x, y) \in \left(\bigcup_{t\in[0,\varepsilon]} X(t) \times Y(0)\right) \bigcup \left(X(0) \times \bigcup_{t\in[0,\varepsilon]} Y(t)\right)$;

(H3) there exists a topology $\mathcal{T}_X$ on X such that for any sequence $\{t_n : 0 < t_n \leqslant \varepsilon\}, t_n \to t_0 = 0$, there exist an $x^0 \in X(0)$ and a subsequence $\{t_{n_k}\}$ of $\{t_n\}$, and for each $k \geqslant 1$, there exists $x_{n_k} \in X(t_{n_k})$ such that (i) $x_{n_k} \to x^0$ in the $\mathcal{T}_X$-topology, and (ii) for all y in $Y(0)$, $\liminf_{t\searrow 0,\ k\to\infty} \partial_t G(t, x_{n_k}, y) \geqslant \partial_t G(0, x^0, y)$;

(H4) there exists a topology $\mathcal{T}_Y$ on Y such that for any sequence $\{t_n : 0 < t_n \leqslant \varepsilon\}, t_n \to t_0 = 0$, there exist $y^0 \in Y(0)$ and a subsequence $\{t_{n_k}\}$ of $\{t_n\}$, and for each $k \geqslant 1$, there exists $y_{n_k} \in Y(t_{n_k})$ such that (i) $y_{n_k} \to y^0$ in the $\mathcal{T}_Y$-topology, and (ii) for all x in $X(0)$, $\limsup_{t\searrow 0,\ k\to\infty} \partial_t G(t, x, y_{n_k}) \leqslant \partial_t G(0, x, y^0)$;

Then, there exists $(x^0, y^0) \in X(0) \times Y(0)$ such that

$$dg(0) = \min_{x\in X(0)} \sup_{y\in Y(0)} \partial_t G(0, x, y) = \partial_t G(0, x^0, y^0) = \sup_{y\in Y(0)} \min_{x\in X(0)} \partial_t G(0, x, y).$$

Thus, (x^0, y^0) is a saddle point of $\partial_t G(0, x, y)$ on $X(0) \times Y(0)$.

References

[1] Flucher, M., Rumpf, M.: Bernoulli's free-boundary problem, qualitative theory and numerical approximation. J. Reine. Angew. Math. 486, 165–204 (1997)

[2] Henrot, A., Shahgholian, H.: Existence of classical solutions to a free boundary problem for the p-Laplace operator, I: the exterior convex case. J. Reine Angew. Math. 521, 85–97 (2000)

[3] Laurain, A., Privat, Y.: On a Bernoulli problem with geometric constraints. ESAIM Control Optim. Calc. Var. 18, 157–180 (2012)

[4] Alt, A., Caffarelli, L.A.: Existence and regularity for a minimum problem with free boundary. J. Reine. Angew. Math. 325, 105–144 (1981)

[5] Friedrichs, K.O.: Über ein minimumproblem für potentialströmungen mit freiem rand. Math. Ann. 109, 60–82 (1934)

[6] Fasano, A.: Some free boundary problems with industrial applications. In: Leugering, G., Engell, S., Griewank, A., Hinze, M., Rannacher, R., Schulz, V., Ulbrich, M., Ulbrich, S. (eds.) Shape Optimization and Free Boundaries. NATO ASI Series (C: Mathematical and Physical Sciences), vol. 380, pp. 113–142. Springer, Dordrecht (1992)

[7] Eppler, K., Harbrecht, H.: Tracking Neumann data for stationary free boundary problems. SIAM J. Control Optim. 48, 2901–2916 (2009)

[8] Eppler, K., Harbrecht, H.: Tracking the Dirichlet data in L^2 is an ill-posed problem. J. Optim. Theory Appl. 145, 17–35 (2010)

[9] Haslinger, J., Ito, K., Kozubek, T., Kunish, K., Peichl, G.H.: On the shape derivative for problems of Bernoulli type. Interfaces Free Bound. 11, 317–330 (2009)

[10] Haslinger, J., Kozubek, T., Kunish, K., Peichl, G.H.: Shape optimization and fictitious domain approach for solving free-boundary value problems of Bernoulli type. Comput. Optim. Appl. 26(3), 231–251 (2003)

[11] Ito, K., Kunisch, K., Peichl, G.H.: Variational approach to shape derivative for a class of Bernoulli problem. J. Math. Anal. Appl. 314(2), 126–149 (2006)

[12] Rabago, J.F.T., Bacani, J.B.: Shape optimization approach to the Bernoulli problem: a Lagrangian formulation. IAENG Int. J. Appl. Math. 47, 417–424 (2017)

[13] Rabago, J.F.T., Bacani, J.B.: Shape optimization approach for solving the Bernoulli problem by tracking the Neumann data: a Lagrangian formulation. Commun. Pur. Appl. Anal. 17, 2683–2702 (2018)

[14] Abda, A.B., Bouchon, F., Peichl, G.H., Sayeh, M., Touzani, R.: A Dirichlet-Neumann cost functional approach for the Bernoulli problem. J. Eng. Math. 81, 157–176 (2013)

[15] Bacani, J.B.: Methods of shape optimization in free boundary problems. PhD thesis, Karl-Franzens-Universität-Graz, Graz, Austria (2013)

[16] Bacani, J.B., Peichl, G.H.: On the first-order shape derivative of the Kohn-Vogelius cost functional of the Bernoulli problem. Abstr. Appl. Anal. 2013, 19–384320 (2013)

[17] Bacani, J.B., Peichl, G.H.: Solving the exterior Bernoulli problem using the shape derivative approach. In: Mohapatra, R., Giri, D., Saxena, P., Srivastava, P. (eds.) Mathematics and Computing 2013. Mathematics & Statistics, vol. 91, pp. 251–269. Springer, New Delhi (2014)

[18] Eppler, K., Harbrecht, H.: On a Kohn-Vogelius like formulation of free boundary problems. Comput. Optim. App. 52, 69–85 (2012)

[19] Rabago, J.F.T., Azegami, H.: An improved shape optimization formulation of the Bernoulli problem by tracking the Neumann data. J. Eng. Math. 117, 1–29 (2019)

[20] Rabago, J.F.T., Azegami, H.: A new energy-gap cost functional cost functional approach for the exterior Bernoulli free boundary problem. Evol. Equ. Control Theory 8(4), 785–824 (2019)

[21] Rabago, J.F.T., Azegami, H.: A second-order shape optimization algorithm for solving the exterior Bernoulli free boundary problem using a new boundary cost functional. Comput. Opti. Appl. 77(1), 251–305 (2020)

[22] Cheng, X.L., Gong, R.F., Han, W., Zheng, X.: A novel coupled complex boundary method for solving inverse source problems. Inverse Problems, 055002 (2014)

[23] Rabago, J.F.T.: On the new coupled complex boundary method in shape optimization framework for solving stationary free boundary problems. Math. Cotrol Relat. Fields (2022). doi:10.3934/mcrf.2022041

[24] Kohn, R., Vogelius, M.: Relaxation of a variational method for impedance computed tomography. Commun. Pure Appl. Math. 40(6), 745–777 (1987)

[25] Bacani, J.B., Rabago, J.F.T.: On the second-order shape derivative of the Kohn-Vogelius objective functional using the velocity method. Int. J. Differ. Equ. 2015, 10–954836 (2015)

[26] Haslinger, J., Kozubek, T., Kunisch, K., Peichl, G.H.: An embedding domain approach for a class of 2-d shape optimization problems: Mathematical analysis. J. Math. Anal. Appl. 209(2), 665–685 (2004)

[27] Haslinger, J., Mäkinen, R.A.E.: Introduction to Shape Optimization: Theory, Approximation, and Computation. SIAM, Philadelphia (2003)

[28] Boulkhemair, A., Chakib, A.: On the uniform Poincaré inequality. Comm. Partial Differential Equations 32, 1439–1447 (2007)

[29] Boulkhemair, A., Nachaoui, A., Chakib, A.: Uniform trace theorem and application to shape optimization. Appl. Comput. Math. 7, 192–205 (2008)

[30] Boulkhemair, A., Nachaoui, A., Chakib, A.: A shape optimization approach for a class of free boundary problems of Bernoulli type. Appl. Math. 58, 205–221 (2013)

[31] Holzleitner, L.: Hausdorff convergence of domains and their boundaries for shape optimal design. Control Cybern. 30(1), 23–44 (2001)

[32] Chenais, D.: On the existence of a solution in a domain identification problem. J. Math. Anal. Appl. 52, 189–219 (1975)

[33] Bacani, J.B.: On the shape gradient and shape hessian of a shape functional subject to Dirichlet and Robin conditions. Appl. Math. Sci. 8(108), 5387–5397 (2014)

[34] Delfour, M.C., Zolésio, J.-P.: Shape sensitivity analysis by min-max differentiability. SIAM J. Control Optim. 26(4), 834–862 (1988)

[35] Delfour, M.C., Zolésio, J.-P.: Shapes and Geometries: Metrics, Analysis, Differential Calculus, And Optimization, 2nd edn. Adv. Des. Control, vol. 22. SIAM, Philadelphia (2011)

[36] Ito, K., Kunisch, K., Peichl, G.H.: Variational approach to shape derivatives. ESAIM Control Optim. Calc. Var. 14, 517–539 (2008)

[37] Dziri, R., Zolésio, J.-R.: Shape-sensitivity analysis for nonlinear heat convection. Appl. Math. Optim. 35, 1–20 (1997)

[38] Harbrecht, H.: A Newton method for Bernoulli's free boundary problem in three dimensions. Computing (2008) 82, 11–30 (2008)

[39] Fiorenza, R.: Hölder and Locally Hölder Continuous Functions, and Open Sets of Class C^k, $C^{k,\lambda}$. Birkhaüser, Basel (2017)

[40] Delfour, M.C., Zolésio, J.-P.: Anatomy of the shape Hessian. Annali di Matematica Pura ed Applicata 159(1), 315–339 (1991)

[41] Ekeland, I., Temam, R.: Convex Analysis and Variational Problems. North-Holland Publishing Co., Amsterdam. Translated from the French, Studies in Mathematics and its Applications, Vol. 1. (1976)

[42] Medková, D.: The Laplace Equation: Boundary Value Problems on Bounded and Unbounded Lipschitz Domains. Springer, Berlin (2018)

[43] Correa, R., Seeger, A.: Directional derivative of a mimimax function. Nonlinear Anal. 9, 13–22 (1985)

[44] Kashiwabara, T., Colciago, C.M., Dedè, L., Quarteroni, A.: Well-posedness, regularity, and convergence analysis of the finite element

approximation of a generalized Robin boundary value problem. SIAM J. Numer. Anal. 53, 105–126 (2015)

[45] Henrot, A., Pierre, M.: Shape Variation and Optimization: A Geometrical Analysis. Tracts in Mathematics, vol. 28. European Mathematical Society, Zürich (2018)

[46] Sokołowski, J., Zolésio, J.-P.: Introduction to Shape Optimization: Shape Sensitivity Analysis. Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg (1992)

[47] Adams, R.A., Fournier, J.J.F.: Sobolev Spaces. Pure and Applied Mathematics, vol. 140. Academic Press, Amsterdam (2003)

[48] Gilbarg, D., Trudinger, N.S.: Elliptic Partial Differential Equations of Second Order. Springer, Berlin, Heidelberg (1988)

[49] Murat, F., Simon, J.: Sur le contrôle par un domaine géométrique. Research report 76015, Univ. Pierre et Marie Curie, Paris (1976)

[50] Tiihonen, T.: Shape optimization and trial methods for free boundary problems. RAIRO Modél. Math. Anal. Numér. 31, 805–825 (1997)

[51] Neuberger, J.W.: Sobolev Gradients and Differential Equations. Springer, Berlin (1997)

[52] Doğan, G., Morin, P., Nochetto, R.H., Verani, M.: Discrete gradient flows for shape optimization and applications. Comput. Methods Appl. Mech. Engrg. 196, 3898–3914 (2007)

[53] Eppler, K., Harbrecht, H.: Efficient treatment of stationary free boundary problems. Appl. Numer. Math. 56, 1326–1339 (2006)

[54] Rabago, J.F.T., Azegami, H.: Shape optimization approach to defect-shape identification with convective boundary condition via partial boundary measurement. Japan J. Indust. Appl. Math. 31(1), 131–176 (2018)

[55] Hecht, F.: New development in FreeFem++. J. Numer. Math. 20, 251–265 (2012)

[56] Morin, P., Nochetto, R.H., Pauletti, M.S., Verani, M.: Adaptive finite element method for shape optimization. ESAIM Control Optim. Calc. Var. 18, 1122–1149 (2012)