

An Improved Shape Optimization Formulation of the Bernoulli Problem by Tracking the Neumann Data

Julius Fergy T. Rabago · Hideyuki Azegami

Received: date / Accepted: date

Abstract We propose a new shape optimization formulation of the Bernoulli problem by tracking the Neumann data. The associated state problem is an equivalent formulation of the Bernoulli problem with a Robin condition. We devise an iterative procedure based on a Lagrangian-like approach to numerically solve the minimization problem. The proposed scheme involves the knowledge of the shape gradient which is established through the minimax formulation. We illustrate the feasibility of the proposed method and highlight its advantage over the classical setting of tracking the Neumann data through several numerical examples.

Keywords Bernoulli problem · Domain perturbation · Free boundary · Lagrangian method · Minimax formulation · Shape optimization · Shape derivative.

1 Introduction

The Bernoulli problem is considered as a prototype of a stationary free boundary problem. It models different physical phenomena such as electrochemical machining [1], potential flow in fluid mechanics [2], tumor growth [3], optimal insulation [4], heat flow [5] and many more. For other industrial applications and further details on the physical background of these type of problems, interested readers may consult [6, 7] and the references therein.

In general, the Bernoulli problem concerns about the problem of finding a connected domain wherein an associated function is harmonic. A part of the boundary is known and the other one is determined by a set of overdetermined boundary conditions for the state. If the free boundary component is strictly exterior to the fixed part of the boundary, the problem is called exterior Bernoulli problem and interior Bernoulli problem otherwise.

J. F. T. Rabago
Graduate School of Informatics, Nagoya University
A4-2 (780) Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: jfrabago@gmail.com

H. Azegami
Graduate School of Informatics, Nagoya University
A4-2 (780) Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: azegami@i.nagoya-u.ac.jp

In this work, we aim to solve the Bernoulli problem under shape optimization settings. We focus on the exterior problem but the same analysis also applies for the interior case.

The exterior Bernoulli free boundary problem can be stated as follows: given a bounded and connected domain $A \subset \mathbb{R}^2$ with a *fixed boundary* $\Gamma := \partial A$ and a constant $\lambda < 0$, one needs to find a bounded connected domain $B \subset \mathbb{R}^2$ with a *free boundary* $\Sigma := \partial B$, containing the closure of A , and an associated state function $u := u(\Omega)$, where $\Omega = B \setminus \bar{A}$, satisfying the system

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

Here, $\partial_{\mathbf{n}} u := \nabla u \cdot \mathbf{n}$ denotes the normal derivative of u and \mathbf{n} is the outward unit normal vector to the free boundary Σ . To write the overdetermined boundary value problem (1) into a shape optimization problem: find (u, Ω) such that

$$J(\Sigma) = \min_{\tilde{\Sigma}} J(\tilde{\Sigma}) = 0, \quad (2)$$

we propose to track the Neumann data in a least-squares sense:

$$J(\Sigma) = \frac{1}{2} \int_{\Sigma} (\partial_{\mathbf{n}} u - \lambda)^2 ds, \quad (3)$$

where the state solution u satisfies, for a fixed $\beta > 0$, the following equivalent form of (1) with a Robin boundary condition:

$$-\Delta u = 0 \text{ in } \Omega, \quad u = 1 \text{ on } \Gamma, \quad \partial_{\mathbf{n}} u + \beta u = \lambda \text{ on } \Sigma. \quad (4)$$

Note that when (u, Ω) is the solution of (1), $J(\Sigma) = 0$ since $\partial_{\mathbf{n}} u = \lambda$ on Σ . On the other hand, when $J(\Sigma) = 0$, we get from (4) the equation $-\beta u = \partial_{\mathbf{n}} u - \lambda = 0$. Since $\beta > 0$, we obtain $u = 0$ on Σ , and thus the overdetermined system (1).

Other reformulations of (1) into shape optimization problems are also possible and have already been extensively studied in previous investigations (see [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]). In [11] (see also [19]), the authors considered a shape optimization formulation of a slightly general problem similar to (1) with $\lambda \in \mathbb{R}$ using the same objective functional (3). However, the state variable u is subject to the pure Dirichlet problem:

$$-\Delta u = 0 \text{ in } \Omega, \quad u = g \text{ on } \Gamma, \quad u = 0 \text{ on } \Sigma, \quad (5)$$

where $g \in H^{3/2}(\Sigma)$.

To numerically solve (1) via formulation (3)–(4), one needs to solve the minimization problem (2). This requires the expression for the *first-order shape derivative* or *shape gradient* of J which, in the case of the problem setting (3) and (5), was established in [11] through *rearrangement method*. This technique introduced in [20] provides a rigorous computation of the shape gradient and allows one to characterize its form without recourse to the chain-rule approach [21], thereby bypassing the computation of the *material* and *shape derivative* of the states. In [19], the shape optimization formulation (3) and (5) of (1) was re-examined by Bacani and the first author. In particular, the shape derivative of J was computed through minimax formulation in the spirit of [22]. Similar to the rearrangement method, the aforementioned strategy in computing the gradient does not involve the shape derivative of the state u as it naturally introduces the use of an adjoint state variable. The rearrangement method, however, has some sort of an advantage over the minimax formulation in computing the shape derivative of J via the problem setting (3) and (5). Particularly, under the very mild $C^{1,1}$ regularity assumption on the boundary of Ω , Haslinger et al. [11] were able to characterize the shape derivative of J by successfully applying the rearrangement method.

On the other hand, the authors in [19] require Ω to be of class $C^{2,1}$ to establish the expression for the shape gradient of J . This higher regularity of the domain, however, permits one to use Hadamard's domain and boundary differentiation formulas and a classical identity to obtain the boundary integral expression for the shape gradient of J . In addition, a $C^{2,1}$ regularity of the boundary of Ω actually guarantees the existence of the shape gradient as it secures sufficient smoothness for the associated state and adjoint state of the problem (see [19, Remark 2]).

As mentioned above, the shape optimization formulation studied in [11] avoids the computation of the material derivative of the states which in turn requires the introduction of an appropriate adjoint state system. The corresponding adjoint state for the formulation setting (3) and (5) (with $g \equiv 1$) of the Bernoulli problem, however, only enjoys one regularity less compared to that of the state variable u satisfying the state equation (5). In fact, for $C^{1,1}$ domain Ω , the state variable u is H^2 regular, while its corresponding adjoint state (here we denote by p) is only in $H^1(\Omega)$. This motivates us to consider (3) with the newly proposed state equation (4). Given this new state equation associated with (3), the corresponding adjoint state now possesses the same regularity with that of u , for each of a particular class of $C^{k,1}$ ($k \geq 1$) domain Ω . In general, for a $C^{k,1}$ domain Ω , the solution $u \in H^1(\Omega)$ to (4) and its corresponding adjoint $p \in H^1(\Omega)$ satisfying (10) are both in $H^{k+1}(\Omega)$. This higher regularity of the adjoint state, combined with a specific choice of the parameter β , actually provides more stability for the iterative scheme we use here to solve the Bernoulli problem (1). This makes the new formulation more practical for numerically solving the said problem.

The main contribution of the present study is the formulation of a novel Lagrangian-like approach for the numerical realization of (1) by means of the shape optimization formulation (2)–(4). We point out that a similar method was recently used in [19] to numerically solve the exterior Bernoulli problem (1) but with the classical state problem formulation (5). We announce in advance that the proposed formulation not only exhibits faster convergence to the optimal solution but also provides stable approximation of the optimal shape. We support these claims with various numerical examples that are reported in the last part of the paper.

The plan for the rest of the paper is as follows. In Section 2, we recall some basic tools from shape optimization which can be used to calculate the shape derivative of the cost functional. Then, we derive the boundary expression for the shape gradient of the cost using the minimax formulation. In Section 3, we give a similar result for the interior Bernoulli problem both for the case of the classical setting and the proposed formulation without proofs. In Section 4, we describe how the computed shape gradient can be utilized to formulate an efficient iterative procedure to numerically solve the Bernoulli problem. In Section 5, we highlight the advantage of the proposed formulation over the classical setting for both of the exterior and interior case by giving out several numerical examples.

2 Shape Derivative of the Cost Functional

We first recall the concept of the *velocity* (or *speed*) method which will be used to calculate the shape derivative of J . Let $\mathcal{D}^k(\mathbb{R}^d, \mathbb{R}^d)$, $d \geq 2$, be the space of k -times continuously differentiable functions with compact support contained in \mathbb{R}^d , and let $\mathbf{V} \in \mathcal{E}^k := C([0, \varepsilon); \mathcal{D}^k(\mathbb{R}^d, \mathbb{R}^d))$, for some integer $k \geq 2$ and a small real number $\varepsilon > 0$. The field $\mathbf{V}(t)(x) = \mathbf{V}(t, x)$, $x \in \mathbb{R}^d$, generates the transformations $T_t(\mathbf{V})(X) := T_t(X) = x(t; X)$, $t \geq 0$, $X \in \mathbb{R}^d$, through the differential equation

$$\frac{d}{dt} x(t; X) = \mathbf{V}(t, x(t; X)), \quad x(0; X) = X, \quad (6)$$

with the initial value X given. We denote the “transformed domain” $T_t(\mathbf{V})(\Omega)$ at $t \geq 0$ by $\Omega_t(\mathbf{V})$, or simply $\Omega_t := T_t(\Omega)$.

For $t \in (0, \varepsilon)$, the transformation T_t is invertible and $T_t, T_t^{-1} \in \mathcal{D}^1(\mathbb{R}^2, \mathbb{R}^2)$ (see, e.g., [16, Lemma 11]). In addition, the quantity $\det DT_t(X)$ is strictly positive, where $DT_t(X)$ is the Jacobian matrix of the transformation $T_t = T_t(\mathbf{V})$ associated with the velocity field \mathbf{V} . Here, and in what follows, the notations $(DT_t)^{-1}$ and $(DT_t)^{-\top}$ denote the inverse and inverse transpose of the Jacobian matrix DT_t , respectively. Furthermore, for convenience, the notations $A(t) = \det DT_t(X)(DT_t^{-1})(DT_t)^{-\top}$ and $w(t) = \det DT_t(X)|(DT_t)^{-\top} \mathbf{n}|$ referred to as the Jacobian matrix of T_t with respect to the boundary $\partial\Omega$ will be used in the paper.

In this study, the evolutions of the domain Ω , which are all contained in a larger set $U \subset \mathbb{R}^2$ (bounded and connected), also of class $C^{2,1}$, are described using time-independent velocity fields such that an admissible deformation field \mathbf{V} forces Γ to remain invariant; that is, we choose

$$\mathbf{V} \in \Theta := \{\mathbf{V} \in C^{k,1}(\bar{\Omega}, \mathbb{R}^2) : \mathbf{V}|_{\Gamma \cup \partial U} = 0\}. \quad (7)$$

The next two lemmas, whose proofs can be found in [21, 23], will be essential to our analysis.

Lemma 1 *For a function $\varphi \in W_{loc}^{1,1}(\mathbb{R}^2)$ and $\mathbf{V} \in \Theta$, the following formulas hold*

- (i) $\nabla(\varphi \circ T_t) = (DT_t)^\top (\nabla \varphi) \circ T_t$,
- (ii) $\frac{d}{dt}(\varphi \circ T_t) = (\nabla \varphi \cdot \mathbf{V}(t)) \circ T_t$,
- (iii) $\frac{d}{dt}(\varphi \circ T_t^{-1}) = -(\nabla \varphi \cdot \mathbf{V}(t)) \circ T_t^{-1}$,
- (iv) $\frac{d}{dt} \det DT_t = [\operatorname{div} \mathbf{V}(t)] \circ T_t \det DT_t$

Lemma 2 *Consider a fixed vector field $\mathbf{V} \in \Theta$ and let $I = (-t_0, t_0)$ with $t_0 > 0$ sufficiently small. Then, the following regularity properties of the transformation T_t hold*

- | | |
|--|--|
| (i) $t \mapsto \det DT_t(X) \in C^1(I, C(\bar{\Omega}))$
(iii) $t \mapsto w(t) \in C^1(I, C(\Sigma))$
(v) $\frac{d}{dt}w(t) _{t=0} = w'(0) = \operatorname{div}_\Sigma \mathbf{V}$ | (ii) $t \mapsto A(t) \in C^1(I, C^1(\bar{\Omega}))$
(iv) $\lim_{t \searrow 0} w(t) = 1$
(vi) $\frac{d}{dt}A(t) _{t=0} = A'(0)$, |
|--|--|

where $A'(0) = (\operatorname{div} \mathbf{V})\mathbf{I}_2 - (D\mathbf{V} + (D\mathbf{V})^\top)$ and the limits defining the derivatives at $t = 0$ exist uniformly in $x \in \bar{\Omega}$.

Now given a functional $J : \Omega \rightarrow \mathbb{R}$, we say that it has a directional *Eulerian derivative* at Ω in the direction \mathbf{V} if the limit

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

exists. In addition, if the map $\mathbf{V} \mapsto dJ(\Omega)[\mathbf{V}]$ is linear and continuous, then J is *shape differentiable* at Ω , and this mapping will be referred to as the *shape gradient* of J at Ω .

Now, looking at the definition of Θ in (7), it suffices to take $k = 1$ and use rearrangement method to compute the expression for the shape gradient of J as done in [11]. However, since we wish to apply the minimax formulation [22] in computing the shape gradient, we take $k = 2$. This in turn will simplify the derivation of the boundary expression for the shape derivative of J as we shall demonstrate in the proof of the following proposition.

Proposition 1 Let Ω be a $C^{2,1}$ bounded domain. Then, the shape derivative of J along a deformation field $\mathbf{V} \in \Theta$ is given by

$$dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} \mathcal{G} \mathbf{V} \cdot \mathbf{n} ds \quad (8)$$

where

$$\begin{aligned} \mathcal{G} = & \nabla u \cdot \nabla p + (\beta u - \lambda) \partial_{\mathbf{n}} p + \beta p \partial_{\mathbf{n}} u + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} \\ & + \kappa \left[(\beta u - \lambda) p + \frac{1}{2} (\partial_{\mathbf{n}} u - \lambda)^2 \right], \end{aligned} \quad (9)$$

and the adjoint state p satisfies the PDE system

$$-\Delta p = 0 \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma, \quad \partial_{\mathbf{n}} p + \beta p = -(\partial_{\mathbf{n}} u - \lambda) \text{ on } \Sigma. \quad (10)$$

If we choose β to be the mean curvature κ of Σ , i.e., $\beta = \kappa$, the kernel \mathcal{G} simplifies to

$$\mathcal{G} = \nabla u \cdot \nabla p - \kappa(u - p) \partial_{\mathbf{n}} u + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} + \frac{\kappa}{2} (\partial_{\mathbf{n}} u - \lambda)^2. \quad (11)$$

In addition, at the shape solution Ω^* of the Bernoulli problem (1) wherein it holds that $\partial_{\mathbf{n}} u = \lambda$ on Σ^* , we have the necessary optimality condition

$$dJ(\Sigma^*)[\mathbf{V}] = 0 \quad \text{for all } \mathbf{V} \in \Theta.$$

Proof As alluded in Introduction, we shall establish the expression for the shape gradient of J through the minimax formulation. To this end, the proof will be accomplished in several steps.

Step 1: Construction of the appropriate functional. We consider the following functional composed of the objective function and the weak formulation of the state system (over the perturbed domain Ω_t) with the introduction of a Lagrange multiplier to penalize the extra constraint on the fixed boundary:

$$G(t, \varphi, \psi) = \int_{\Omega_t} \nabla \varphi \cdot \nabla \psi dx_t + \int_{\Sigma_t} \left[(\beta \varphi - \lambda) \psi + \frac{1}{2} (\partial_{\mathbf{n}} \varphi - \lambda)^2 \right] ds_t + \int_{\Gamma_t} (\varphi - 1) \partial_{\mathbf{n}} \psi ds_t.$$

In above expression for G , one can actually drop t in Γ_t since Γ is invariant during deformation (i.e., $\Gamma_t = \Gamma$ for all t).

One can easily check that, at $t = 0$,

$$J(\Sigma) = \min_{\varphi \in H^1(\Omega)} \max_{\psi \in V(\Omega)} G(0, \varphi, \psi)$$

since

$$\max_{\psi \in V(\Omega)} G(0, \varphi, \psi) = \begin{cases} \frac{1}{2} \int_{\Sigma} (\partial_{\mathbf{n}} u - \lambda)^2 ds & \text{if } \varphi = u, \\ +\infty & \text{otherwise.} \end{cases}$$

In addition, one can also show that the functional $G(0, \varphi, \psi)$ is convex continuous with respect to φ and concave continuous with respect to ψ . Hence, according to [24], the functional admits a saddle point (u, p) provided that the pair (u, p) satisfies the variational forms of the systems (4) and (10):

$$\int_{\Omega} \nabla u \cdot \nabla \psi dx + \beta \int_{\Sigma} u \psi ds = \int_{\Sigma} \lambda \psi ds, \quad \forall \psi \in V(\Omega), \quad u \in H^1(\Omega), \quad u|_{\Gamma} = 1, \quad (12)$$

$$\int_{\Omega} \nabla p \cdot \nabla \varphi dx + \beta \int_{\Sigma} p \varphi ds = - \int_{\Sigma} (\partial_{\mathbf{n}} u - \lambda) \varphi ds, \quad \forall \varphi \in V(\Omega), \quad p \in V(\Omega). \quad (13)$$

Here, the space of test functions $V(\Omega)$ is given by the Hilbert space

$$V(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma} = 0\}$$

endowed with the norm

$$\|v\|_{V(\Omega)}^2 = \int_{\Omega} |\nabla v|^2 dx + \int_{\Sigma} |v|^2 ds.$$

The said saddle point is unique due to the unique solvability of (12) and (13). A similar analysis also holds on the transformed domain Ω_t . In fact, we have the equality

$$J(\Sigma_t) = \min_{\varphi \in H^1(\Omega_t)} \max_{\psi \in V(\Omega_t)} G(t, \varphi, \psi). \quad (14)$$

The corresponding saddle point of $G(t, \varphi, \psi)$, (u_t, p_t) , for non-zero small t , is characterized by the same weak forms (12) and (13), only that the integrals are defined over Ω_t with test functions from $V(\Omega_t)$.

Step 2: Getting rid of the time-dependence of the function spaces. Our aim is to get the derivative of the minimax functional $G(t, \varphi, \psi)$ with respect to the parameter $t \geq 0$ through the application of Theorem 1 due to Correa and Seeger [25] (see Appendix). However, the function spaces appearing in the minimax in (14) depend on the parameter t . To get around this difficulty, we make use of the *function space parametrization technique* put forward in [21]. That is, we parametrize the functions in $H^1(\Omega_t)$ (resp. $V(\Omega_t)$) by elements of $H^1(\Omega)$ (resp. $V(\Omega)$) using the map

$$\varphi \mapsto \varphi \circ T_t^{-1} : H^1(\Omega) \rightarrow H^1(\Omega_t) \quad (\text{resp. } V(\Omega) \rightarrow V(\Omega_t)).$$

This parametrization does not change the values of the saddle points. Thus, we have a new functional $G(t, \varphi \circ T_t^{-1}, \psi \circ T_t^{-1})$ with the same saddle point for $G(t, \varphi, \psi)$. We rewrite the resulting functional and the systems characterizing its saddle points into their respective equivalent forms via domain and boundary transformations. Particularly, we have

$$\begin{aligned} \tilde{G}(t, \varphi, \psi) := & \int_{\Omega} A(t) \nabla \varphi \cdot \nabla \psi dx + \int_{\Gamma} (\varphi - 1) \partial_n \psi ds \\ & + \int_{\Sigma} w(t) \left[(\beta \varphi - \lambda) \psi + \frac{1}{2} (\partial_n \varphi - \lambda)^2 \right] ds, \end{aligned}$$

where the saddle point $(u^t, p^t) \in H^1(\Omega) \times V(\Omega)$ of this new functional is characterized by the systems

$$\int_{\Omega} A(t) \nabla u^t \cdot \nabla \psi dx + \beta \int_{\Sigma} w(t) u^t \psi ds = \int_{\Sigma} w(t) \lambda \psi ds, \quad \forall \psi \in V(\Omega), \quad u^t|_{\Gamma} = 1, \quad (15)$$

$$\int_{\Omega} A(t) \nabla p^t \cdot \nabla \varphi dx + \beta \int_{\Sigma} w(t) p^t \varphi ds = - \int_{\Sigma} w(t) (\partial_n u^t - \lambda) \varphi ds, \quad \forall \varphi \in V(\Omega). \quad (16)$$

In the expression for $\tilde{G}(t, \varphi, \psi)$, we have used the fact that $\text{const.} \circ T_t = \text{const.}$, and $T_t(x) = x$ and $w(t) = 1$ on $\Gamma_t = \Gamma$.

Step 3: Verifying the four assumptions of Theorem 1. To get the shape derivative of J along a deformation field \mathbf{V} , we evaluate the limit

$$\lim_{t \searrow 0} \left(\min_{\varphi \in H^1(\Omega)} \max_{\psi \in V(\Omega)} \frac{G(t, \varphi \circ T_t^{-1}, \psi \circ T_t^{-1}) - G(0, \varphi, \psi)}{t} \right).$$

To do this, we apply Theorem 1. But first, we need to verify its four assumptions.

We let $\mathbf{V} \in \Theta$ and choose a sufficiently small number $\varepsilon > 0$, such that, for all $t \in [0, \varepsilon]$, there exist some constants $\alpha_1, \alpha_2, \beta_1$ and β_2 satisfying $0 < \alpha_1 \leq \alpha_2, 0 < \beta_1 \leq \beta_2$ and such that $\alpha_1|\xi|^2 \leq A(t)\xi \cdot \xi \leq \alpha_2|\xi|^2$, for all $\xi \in \mathbb{R}^2$ and $\beta_1 \leq w(t) \leq \beta_2$ (cf. [16]). We define the sets

$$\begin{aligned} X(t) &:= \left\{ x^t \in H^1(\Omega) : \sup_{y \in V(\Omega)} \tilde{G}(t, x^t, y) = \inf_{x \in H^1(\Omega)} \sup_{y \in V(\Omega)} \tilde{G}(t, x, y) \right\}, \\ Y(t) &:= \left\{ y^t \in V(\Omega) : \inf_{x \in H^1(\Omega)} \tilde{G}(t, x, y^t) = \sup_{y \in V(\Omega)} \inf_{x \in H^1(\Omega)} \tilde{G}(t, x, y) \right\}. \end{aligned}$$

The functions u^t and p^t satisfy the inequality $G(t, u^t, \psi) \leq G(t, u^t, p^t) \leq G(t, \varphi, p^t)$. Hence, it is evident that $X(t)$ and $Y(t)$ are non-empty, since, in particular, we have $X(t) = \{u^t\}$ and $Y(t) = \{p^t\}$. Thus, we get

$$\forall t \in [0, \varepsilon] : S(t) := X(t) \times Y(t) = \{u^t, p^t\} \neq \emptyset.$$

This shows that condition (H1) is satisfied.

To verify condition (H2), we compute the derivative of $\tilde{G}(t, \varphi, \psi)$ with respect to $t \geq 0$:

$$\partial_t \tilde{G}(t, \varphi, \psi) = \int_{\Omega} A'(t) \nabla \varphi \cdot \nabla \psi \, dx + \int_{\Sigma} w'(t) \left[(\beta \varphi - \lambda) \psi + \frac{1}{2} (\partial_{\mathbf{n}} \varphi - \lambda)^2 \right] \, ds.$$

Since $\mathbf{V} \in \mathcal{D}^1(\mathbb{R}^2, \mathbb{R}^2)$ and the maps $t \mapsto DT_t$ are continuous in $[0, \varepsilon]$ (see Lemma 2), the partial derivative $\partial_t \tilde{G}(t, \varphi, \psi)$ exists everywhere in $[0, \varepsilon]$. Hence, (H2) is satisfied.

To check (H3)(i) and (H4)(i), we first show the boundedness of (u^t, p^t) . We take $\psi = u^t$ in (15). With the choice of ε , we can use the bounds for $A(t)$ and $w(t)$ to get the estimate

$$\min\{\alpha_1, \beta\beta_1\} \|u^t\|_{V(\Omega)}^2 \leq \alpha_2 |\lambda| |\Sigma|^{1/2} \|u^t\|_{L^2(\Sigma)}.$$

Since the norm $\|\cdot\|_{V(\Omega)}$ is equivalent to the usual H^1 Sobolev norm, there exist some constants $c_1, c_2 > 0$ such

$$\|u^t\|_{H^1(\Omega)} \leq \frac{c_2 \alpha_2}{c_1 \min\{\alpha_1, \beta\beta_1\}} |\lambda| |\Sigma|^{1/2}.$$

Applying the same technique, we can also show that p^t is bounded.

Next we show the continuity of the pair (u^t, p^t) . To prove the continuity of u^t , we subtract in (15) at $t > 0, t = 0$ and let $\psi = u^t - u$ to obtain

$$\begin{aligned} \|u^t - u\|_{V(\Omega)}^2 &= \int_{\Omega} (A(t) - \mathbf{I}_2) \nabla u^t \cdot \nabla (u^t - u) \, dx + \int_{\Sigma} (\lambda \circ T_t w(t) - \lambda) (u^t - u) \, ds \\ &\quad - \beta \int_{\Sigma} u^t (w(t) - 1) (u^t - u) \, ds \\ &\leq |A(t) - \mathbf{I}_2| \|u^t\|_{H^1(\Omega)} \|u^t - u\|_{L^2(\Omega)} + \|\lambda \circ T_t w(t) - \lambda\|_{L^2(\Sigma)} \|u^t - u\|_{L^2(\Sigma)} \\ &\quad + \beta |w(t) - 1| \|u^t\|_{L^2(\Sigma)} \|u^t - u\|_{L^2(\Omega)}. \end{aligned}$$

Using the boundedness of u^t and the equivalence of the norms $\|\cdot\|_{V(\Omega)}$ and $\|\cdot\|_{H^1(\Omega)}$, we get the bound

$$\|u^t - u\|_{H^1(\Omega)} \leq c_3 \left(|A(t) - \mathbf{I}_2| + \|\lambda \circ T_t w(t) - \lambda\|_{L^2(\Sigma)} + |w(t) - 1| \right)$$

for some constant $c_3 > 0$. Hence, $u^t \rightarrow u$ in $H^1(\Omega)$ because $A(t) - \mathbf{I}_2 \rightarrow 0$, $w(t) \rightarrow 1$ and $\lambda \circ T_t \rightarrow \lambda$ as $t \rightarrow 0$ (Lemma 2). With the use of a classical regularity theorem [26] and standard arguments, we can show that u^t is also bounded in $H^2(\Omega)$ since u^t is in $H^2(\Omega)$. This implies the continuity of u^t in $H^2(\Omega)$, and thus verifies (H3)(i) for $H^2(\Omega)$ -strong. Using a similar argument, we can also show that $p^t \rightarrow p$ strongly in $H^1(\Omega)$ as $t \rightarrow 0$. Moreover, since $u^t \in H^2(\Omega)$, $\partial_{\mathbf{n}} u^t - \lambda \in H^{1/2}(\Sigma)$. By regularity theorem, p^t is also in $H^2(\Omega)$ and so the continuity of p^t in $H^2(\Omega)$. Hence, condition (H4)(i) is true for the $H^2(\Omega)$ -strong topology.

Finally, conditions (H3)(ii) and (H4)(ii) are easily verified by the strong continuity of the maps $(t, \varphi) \mapsto \partial_t \tilde{G}(t, \varphi, \psi)$ and $(t, \psi) \mapsto \partial_t \tilde{G}(t, \varphi, \psi)$.

Consequently, we have verified all assumptions of Theorem 1 from which we conclude that following equation holds

$$\begin{aligned} dJ(\Sigma)[\mathbf{V}] &= \partial_t \tilde{G}(t, u, p)|_{t=0} \\ &= \int_{\Omega} A'(0) \nabla u \cdot \nabla p \, dx + \int_{\Sigma} w'(0) \left[(\beta u - \lambda) p + \frac{1}{2} (\partial_{\mathbf{n}} u - \lambda)^2 \right] ds, \end{aligned} \quad (17)$$

where $A'(0) = (\operatorname{div} \mathbf{V}) \mathbf{I}_2 - (D\mathbf{V} + (D\mathbf{V})^\top)$ and $w'(0) = \operatorname{div}_{\Sigma} \mathbf{V}$ (see Lemma 2), and $(u, p) \in H^1(\Omega) \times H^1(\Omega)$ is the unique solution pair to systems (4) and (10).

Step 4: Characterization of the shape gradient in terms of just a boundary expression.
It can be checked without difficulty that the map $\mathbf{V} \mapsto dJ(\Sigma)[\mathbf{V}] : \mathcal{D}^1(\mathbb{R}^2, \mathbb{R}^2) \rightarrow \mathbb{R}$ is linear and continuous. Then, according to Hadamard-Zolésio structure theorem [21], there exists, for a $C^{2,1}$ domain Ω , a scalar distribution $\mathcal{G}(\Sigma) \in \mathcal{D}^1(\Sigma)$ such that $dJ(\Sigma)[\mathbf{V}] = \langle \mathcal{G}(\Sigma), \mathbf{V} \cdot \mathbf{n} \rangle$.

Now we further characterize this boundary expression as follows. Firstly, we note that for a $C^{2,1}$ domain Ω , the unique solution pair to (4) and (10) possesses more regularity. In fact, u and p are elements of $H^3(\Omega)$ for Ω of class $C^{2,1}$ (cf. [27]). This aforementioned regularity allows us to use Hadamard's domain and boundary differentiation formulas (cf. [21, Thm. 4.3, p. 486]):

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_t} f(t, x) \, dx_t &= \int_{\Omega_t} \frac{\partial f}{\partial t}(t, x) \, dx_t + \int_{\partial\Omega_t} f(t, x) \mathbf{V}(t) \cdot \mathbf{n} \, ds_t \\ \frac{d}{dt} \int_{\partial\Omega_t} f(t, s) \, ds_t &= \int_{\partial\Omega_t} \frac{\partial f}{\partial t}(t, s) \, ds_t + \int_{\partial\Omega_t} \left(\frac{\partial f}{\partial \mathbf{n}}(t, s) + \kappa f(t, s) \right) \mathbf{V}(t) \cdot \mathbf{n} \, ds_t \end{aligned}$$

to evaluate the partial derivative $\partial_t G(t, \varphi \circ T_t^{-1}, \psi \circ T_t^{-1})$ at $t = 0$. That is, we have

$$\begin{aligned} \partial_t G(t, \varphi \circ T_t^{-1}, \psi \circ T_t^{-1})|_{t=0} &= \int_{\Omega} (\nabla \dot{\varphi} \cdot \nabla \psi + \nabla \varphi \cdot \nabla \dot{\psi}) \, dx + \int_{\Sigma} \left[(\beta \varphi - \lambda) \dot{\psi} + \beta \dot{\varphi} \psi + (\partial_{\mathbf{n}} \varphi - \lambda) \partial_{\mathbf{n}} \dot{\psi} \right] ds \\ &\quad + \int_{\Sigma} \nabla \varphi \cdot \nabla \psi \mathbf{V} \cdot \mathbf{n} \, ds + \int_{\Sigma} \partial_{\mathbf{n}} \left[(\beta \varphi - \lambda) \psi + \frac{1}{2} (\partial_{\mathbf{n}} \varphi - \lambda)^2 \right] \mathbf{V} \cdot \mathbf{n} \, ds \\ &\quad + \int_{\Sigma} \kappa \left[(\beta \varphi - \lambda) \psi + \frac{1}{2} (\partial_{\mathbf{n}} \varphi - \lambda)^2 \right] ds, \end{aligned}$$

where (see Lemma 1)

$$\dot{\varphi} = \frac{d}{dt} \varphi \circ T_t^{-1} \Big|_{t=0} = -\nabla \varphi \cdot \mathbf{V} \in H^1(\Omega), \quad \dot{\psi} = \frac{d}{dt} \psi \circ T_t^{-1} \Big|_{t=0} = -\nabla \psi \cdot \mathbf{V} \in H^1(\Omega).$$

Substituting (φ, ψ) by (u, p) , we see that the first and second integral vanish due to (12) with $\psi = -\nabla p \cdot \mathbf{V}$ and (13) with $\varphi = -\nabla u \cdot \mathbf{V}$. Accordingly, we get

$$dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} \mathcal{G} \mathbf{V} \cdot \mathbf{n} ds$$

where

$$\mathcal{G} = \nabla u \cdot \nabla p + (\beta u - \lambda) \partial_{\mathbf{n}} p + \beta p \partial_{\mathbf{n}} u + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} + \kappa \left[(\beta u - \lambda) p + \frac{1}{2} (\partial_{\mathbf{n}} u - \lambda)^2 \right],$$

as desired.

If we take $\beta = \kappa$, (9) simplifies to

$$\mathcal{G} = \nabla u \cdot \nabla p - \kappa(u - p) \partial_{\mathbf{n}} u + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} + \frac{\kappa}{2} (\partial_{\mathbf{n}} u - \lambda)^2.$$

Moreover, if Ω^* is such that $u = u(\Omega^*)$ is the solution to the Bernoulli problem (1), i.e., it holds that $\partial_{\mathbf{n}} u = \lambda$ on Σ^* , then $p = p(\Omega^*)$ is identically equal to zero. Hence, $\mathcal{G} = 0$ on Σ^* , and this implies that $dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} (0) \mathbf{V} \cdot \mathbf{n} ds = 0$.

Evidently, the computed shape gradient of J under the proposed formulation differs from the classical one (see [11, Thm. 3.1]). We recall that the cost function J with state variable u satisfying (5) (with $g \equiv 1$) has the shape derivative given by

$$dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} \mathcal{G}_0 \mathbf{V} \cdot \mathbf{n} ds := - \int_{\Sigma} \left[\partial_{\mathbf{n}} u \partial_{\mathbf{n}} p + \kappa \left(\frac{1}{2} p^2 + \lambda p \right) \right] \mathbf{V} \cdot \mathbf{n} ds,$$

where the adjoint state p satisfies

$$-\Delta p = 0 \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma, \quad p = \partial_{\mathbf{n}} u - \lambda \text{ on } \Sigma. \quad (18)$$

3 Interior Bernoulli Problem

Given a bounded domain $A \subset \mathbb{R}^2$ with boundary Γ and a constant $\lambda > 0$, the interior Bernoulli problem consists in finding a bounded domain $B \subset \bar{A}$ with boundary Σ and a function u defined on $\Omega = A \setminus \bar{B}$ such that

$$-\Delta u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma, \quad u = 1 \quad \text{and} \quad \partial_{\mathbf{n}} u = \lambda \text{ on } \Sigma, \quad (19)$$

where \mathbf{n} is the interior unit normal to Σ .

The interior Bernoulli problem (19) can be rephrased into various shape optimization setting and one way to do this is to track the Neumann data similar to that of [11] for the exterior case. More precisely, one could consider the minimization problem (2) where the cost function J is given by (3) but with state constraint

$$-\Delta u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma, \quad u = 1 \text{ on } \Sigma. \quad (20)$$

In this case, the shape derivative of J is given as follows.

Proposition 2 Let Ω be a $C^{2,1}$ bounded domain. Then, the shape derivative of the cost function $J(\Sigma)$ (subject to (20)) along a deformation field $\mathbf{V} \in \Theta$ is given by

$$dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} \mathcal{G}_1 \mathbf{V} \cdot \mathbf{n} ds,$$

where

$$\mathcal{G}_1 = \nabla u \cdot \nabla p + \partial_{\mathbf{n}} u \partial_{\mathbf{n}} p + p \partial_{\mathbf{n}} u + (u-1) \frac{\partial^2 p}{\partial \mathbf{n}^2} + \kappa \left[(u-1) \partial_{\mathbf{n}} p + \frac{1}{2} p^2 \right],$$

and p is the corresponding adjoint state variable satisfying the same system (18).

Obviously, one could also consider a reformulation of (19) similar to our proposed method by constructing an associated state problem with Robin condition. Instead of (18), one may opt to use the state equation

$$-\Delta u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma, \quad \partial_{\mathbf{n}} u + \beta u = \lambda + \beta \text{ on } \Sigma, \quad (21)$$

where $\beta > 0$. With this new state equation, the shape gradient of J has now a different structure as evident in the following result.

Proposition 3 Let Ω be a $C^{2,1}$ bounded domain. Then, the shape derivative of the cost function $J(\Sigma)$ (subject to (21)) along a deformation field $\mathbf{V} \in \Theta$ is given by

$$dJ(\Sigma)[\mathbf{V}] = \int_{\Sigma} \mathcal{G}_2 \mathbf{V} \cdot \mathbf{n} ds,$$

where

$$\begin{aligned} \mathcal{G}_2 = & \nabla u \cdot \nabla p + \partial_{\mathbf{n}} p (\beta u - \lambda - \beta) + \beta p \partial_{\mathbf{n}} u + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} \\ & + \kappa \left[p(\beta u - \lambda - \beta) + \frac{1}{2} (\partial_{\mathbf{n}} u - \lambda)^2 \right], \end{aligned}$$

and p denotes the adjoint state satisfying the same system (10).

If $\beta = \kappa$, the kernel \mathcal{G}_2 simplifies to

$$\mathcal{G}_2 = \nabla u \cdot \nabla p - \partial_{\mathbf{n}} u \partial_{\mathbf{n}} p + (\partial_{\mathbf{n}} u - \lambda) \frac{\partial^2 u}{\partial \mathbf{n}^2} + \frac{\kappa}{2} (\partial_{\mathbf{n}} u - \lambda)^2.$$

We omit the proofs of these two propositions since they are similar to that of Proposition 1.

4 Numerical Approximation

To the best of our knowledge, there are at least three different computational strategies for the numerical resolution of the Bernoulli problem (1). The first one is the *fixed-point* approach wherein a sequence of elliptic problems are solved in a sequence of converging domains with one of the conditions on the free boundary omitted, and then the remaining boundary condition is used to update the free boundary (see [28,29]). This approach does not require any gradient information in contrast to the second approach which considers an equivalent shape optimization formulation of the problem (see, e.g., [9,10,11,15,18,19]). Another strategy, built from the theory of complex analysis, is the use of *conformal mapping* method. This solution method was recently developed by Haddar and Kress in [30] which relates the Bernoulli problem in the context of inverse problems. In a more recent study, another method was also introduced by Kress in [31] in an attempt to improve the use of boundary integral equations for numerically solving the Bernoulli problem. In terms

of numerical performance, it was revealed in [31] that this recently proposed method inspired by Trefftz' integral equation method [32] is more robust and wider applicable than that of [30]. We mention that Trefftz' approach, in principle, can be considered as a so-called *trial method* which is also a prominent numerical method for solving free boundary value problems such as the Bernoulli problem (see [8, 33, 34, 35]).

In this paper, we use a classical gradient scheme for the numerical realization of the proposed method by means of a Lagrangian-like method. It consists in adopting an iterative procedure that decreases the value of the cost functional J at each iteration. One could also use an Eulerian-like approach such as the level-set method that was applied, for instance, in [15, 20]. Alternatively, one could also apply a variant of Newton's method to numerically solve the minimization problem. This method, however, also requires the knowledge of the shape Hessian of J which is considerably more difficult to obtain and utilize (see, e.g., [36, 37], and the references cited therein).

4.1 A Gradient-type Algorithm

Let us denote by Ω_k and Σ_k the shapes of the domain and the free (exterior) boundary at the k^{th} iteration, respectively. A descent direction for the algorithm can be found by defining

$$\mathbf{V} = -\mathcal{G}\mathbf{n}. \quad (22)$$

and then we can update the shape Ω as $\Omega_{k+1} := \Omega_{t_{k+1}} = (\mathbf{I}_2 + t_k \mathbf{V})\Omega$, where the step size parameter t_k is such that $t_k \in (0, \varepsilon]$ for some small real number $\varepsilon > 0$. However, direct application of (22) may cause oscillations on the boundary of the approximate solution. To avoid such phenomena, we take the descent direction $\mathbf{V} \in [V(\Omega)]^2$ as the unique solution of the variational problem

$$\int_{\Omega} \nabla \mathbf{V} : \nabla \varphi \, dx + \int_{\Sigma} \beta(\mathbf{V} \cdot \mathbf{n}) \mathbf{n} \cdot \varphi \, ds = - \int_{\Sigma} \mathcal{G}\mathbf{n} \cdot \varphi \, ds, \quad \forall \varphi \in [V(\Omega)]^2. \quad (23)$$

The resulting vector field \mathbf{V} (also known in the literature as Sobolev gradient [38]) obtained through (23) now provides a smooth extension of $\mathcal{G}\mathbf{n}$ over the entire domain Ω which not only smoothes the boundary (see [39, 40]) but also preconditions the descent direction. Equation (23) is actually a variant of the so-called H^1 gradient method [41] which, on the other hand, was inspired by the idea of the *traction method* [39, 42, 43, 44].

The main steps required for the computation of the k^{th} domain are given as follows:

Algorithm

- Step 1. Choose an initial shape Ω_0 ;
- Step 2. compute the solutions u and p of problems (4) and (10) on Ω_k ;
- Step 3. evaluate the descent direction \mathbf{V}_k using (23);
- Step 4. set $\Omega_{k+1} = (\mathbf{I}_2 + t_k \mathbf{V}_k)\Omega_k$, where t_k is a positive scalar.

Note that, in view of (9) and (23), the computation of the descent direction \mathbf{V} in Step 3 demands the evaluation of the mean curvature κ of Σ . We recall from [45, Prop. 5.4.8, p. 218] that, for a domain Ω of C^2 class, the mean curvature can be defined as

$$\kappa = \operatorname{div}_{\Sigma} \mathbf{n} = \operatorname{div} \mathbf{N},$$

where \mathbf{N} is any (unitary) extension of \mathbf{n} that is of class C^1 . Following this idea, we can therefore calculate κ by evaluating the expression $\operatorname{div} \mathbf{N}$, where \mathbf{N} is the unique element in $[H^1(\Omega)]^2$ of the variational equation

$$\int_{\Omega} \nabla \mathbf{N} : \nabla \varphi \, dx + \int_{\Sigma} \mathbf{N} \cdot \varphi \, ds = \int_{\Sigma} \mathbf{n} \cdot \varphi \, ds, \quad \forall \varphi \in [H^1(\Omega)]^2.$$

Furthermore, notice in (9) that the kernel \mathcal{G} also involves the computation of the second-derivative $\partial^2 u / \partial \mathbf{n}^2$. In the classical approach (formulation (3) and (5)), this can be simplified as $-\partial^2 u / \partial \mathbf{n}^2 = \kappa \partial u / \partial \mathbf{n}$ since $\Delta u = \Delta_\Sigma u + \kappa \partial u / \partial \mathbf{n} + \partial^2 u / \partial \mathbf{n}^2$ and $u|_\Sigma = 0$ (see equation (5)). However, in our proposed numerical procedure, we calculate $\partial^2 u / \partial \mathbf{n}^2$ using a similar idea in computing κ ; i.e., we evaluate $\partial^2 u / \partial \mathbf{n}^2$ by computing the normal derivative of a smooth extension of $\partial_{\mathbf{n}} u$. We mention here that, to the best of our knowledge, such method for numerically computing a second order normal derivative is also novel to our work.

4.2 Step Size

The choice of the step size parameter t_k is not an easy task. Too large, the algorithm is unstable; too small, the rate of convergence is insignificant. In updating $t_k \in (0, \varepsilon]$, where $\varepsilon > 0$ is some sufficiently small real number, the following heuristic which is inspired by the Armijo-Goldstein line search strategy (cf. [10]) will be employed. In view of (22) and the definition of the domain Ω_ε , we have $J(\Sigma_\varepsilon) \simeq J(\Sigma_0) + \varepsilon dJ(\Sigma_0)[\mathbf{V}] = J(\Sigma_0) - \varepsilon \|\mathcal{G}\|_{L^2(\Sigma_0)}^2$. The requirement $J(\Sigma_\varepsilon) = (1 - \alpha)J(\Sigma_0)$ for some $\alpha \in (0, 1)$ then suggests the choice $\varepsilon = \alpha J(\Sigma_0) / \|\mathcal{G}\|_{L^2(\Sigma_0)}^2$. However, since we are regularizing in Step 3 the descent direction \mathbf{V} for the present algorithm using equation (23), we replace the L^2 -norm of \mathcal{G} appearing in the denominator of the previous formula with $\|\mathbf{V}\|_{\mathcal{X}^2}^2$, and then finally define the step size t_k as

$$t_k = \alpha J(\Sigma_k) / \|\mathbf{V}\|_{\mathcal{X}^2}^2. \quad (24)$$

Here, the function space \mathcal{X} is either the space $H^1(\Omega_k)$, $V(\Omega_k)$ or $L^2(\Sigma_k)$.

We further explain the above formula as follows. First we note that, in general, we could regularize the descent direction \mathbf{V} using the variational equation

$$\mathcal{W}(\mathbf{V}, \varphi) = -\langle G, \varphi \rangle, \quad \forall \varphi \in [V(\Omega)]^2, \quad (25)$$

where $\mathcal{W}(\cdot, \cdot)$ is some bounded coercive bilinear form on an appropriate space \mathcal{X} (see, e.g., [39, Sec. 6.3]). Then, from (25) and the requirement that the relation

$$J(\Sigma_\varepsilon) = (1 - \alpha)J(\Sigma_0) = J(\Sigma_0) + \varepsilon \langle G, \mathbf{V} \rangle$$

holds for some $\alpha \in (0, 1)$, we get the equation $\varepsilon = -\alpha J(\Sigma_0) / \langle G, \mathbf{V} \rangle = \alpha J(\Sigma_0) / \mathcal{W}(\mathbf{V}, \mathbf{V})$, for any $\mathbf{V} \in [V(\Omega)]^2$. Hence, at each iteration, we may choose, for a fixed α , the step size parameter t_k as

$$t_k = \alpha J(\Sigma_k) / \mathcal{W}(\mathbf{V}, \mathbf{V}).$$

The above formula for t_k clearly provides a natural choice for the magnitude of the step size when the descent direction \mathbf{V} is regularized through equation (25). However, we alter here the choice of norm in (24) (while using (23) for preconditioning the descent direction) so as not to make the ratio between the cost and the norm of the descent direction \mathbf{V} very small.

Now, with $\alpha \in (0, 1)$ fixed, the step size will be decided in the following fashion: we take t_k as in (24) whenever there is a decrease in the computed cost value from the previous to the next iteration loop (i.e., if $J(\Sigma_{k+1}) \leq J(\Sigma_k)$). Otherwise, if the cost value increases, we reduce the step size (exactly by half) and go backward: the next iteration is initialized with the previous shape Ω_k . We also reduce the step size t_k if reversed triangles are detected within the mesh update.

4.3 Stopping Criterion

A typical stopping criterion for gradient-type algorithm is to find that whether the shape gradients in some suitable norm are small enough. However, since we use the continuous shape gradients, it is hopeless for us to expect very small gradient norm because of numerical discretization errors. Another option, however, is to use the inequality condition

$$|J(\Sigma_{k+1}) - J(\Sigma_k)| < \text{Tol} \quad (26)$$

as a stopping rule, where $\text{Tol} > 0$ is a predetermined small tolerance value. Even so, because we want to compare our proposed formulation with that of the classical Neumann-data tracking approach, the proper choice for Tol in (26) may be different for each of the two formulations. Because of these issue, we need to consider a stopping rule that is independent of the cost or the gradient value. In this regard, the most reasonable choice would be to use the computing time (i.e., the maximum number of seconds before timing out) as the stopping condition. Note that we could also stop the iteration process with a predetermined maximum number of iterations. However, since the step size t_k is chosen on the basis of formula (24), the total number of iterations the algorithm needed to process in order to obtain good enough approximation of the optimal shape solution (given that α and X are fixed) may differ greatly from each formulation. Nevertheless, we emphasize that we could still utilize the inequality condition given in (26) to choose an optimal iteration number that provides reasonable approximation of the optimal solution. This can be done by first running the algorithm for a certain amount of time and then examine afterwards the convergence history of the cost function (or possibly the history of Hausdorff distances between the k^{th} and final computed shape) to decide for the best choice of the tolerance value. The index k that satisfies the condition (26) with the chosen value for Tol can then be regarded as the optimal iteration number when the said stopping rule is applied.

To end this section, we mention that we also apply the same algorithm presented above when using the Neumann-data tracking approach with only a few modifications. We replace in Step 2 of the algorithm the adjoint state problem (10) with the PDE system (18) and the descent direction is computed with \mathcal{G} replaced by \mathcal{G}_0 .

5 Numerical Tests

Now we illustrate the feasibility of the proposed algorithm through various numerical examples. At this juncture, we mention that the existence of solutions for the exterior problem can be established by means of sub- and supersolutions [46] or through variational methods in the context of shape optimization [47]. The question of uniqueness of solutions, on the other hand, can be guaranteed for convex domains in the case of exterior problems (see [48]) which is not true, however, for the interior problem. In fact, the interior case need not have a solution for every domain A and for every positive constant $\lambda > 0$. Nevertheless, at least one solution exists for the more general case of p -Laplacian when A is a convex domain with smooth (at least C^1) boundary and λ is not less than the *Bernoulli constant* $\lambda^*(\Omega) > 0$ (see [49]). Still, uniqueness of solution holds for the interior case when $\lambda = \lambda^*(\Omega)$ [50].

The numerical simulations are carried out in two-dimension using the programming software FREEFEM++ (see [51]). We use a P2 finite element discretization to solve the state problem (4), the adjoint state problem (10) and all other variational equations involve in the iterative procedure. In all numerical experiments conducted here, the number of discretization points on the free and fixed boundaries are initially set to $N_{\text{ext}} = 120$ and $N_{\text{int}} = 100$,

respectively. Meanwhile, we utilize the function *movemesh* of FREEFEM++ in deforming the shape of the domain at each step and use the function *adaptmesh* to refine and avoid the degeneracy of the triangles in the meshes. During mesh adaptation, the maximum edge size of the mesh is taken equal to h_{\max} . All computations are carried out on a 1.6 GHz Intel Core i5 Macintosh computer with 4GB RAM processors.

In addition to the above specifications, we take β as the mean curvature of the free boundary Σ (i.e., we let $\beta = \kappa$) in all of the test cases we examine here. We mention that we have actually tested several values for this Robin coefficient, but it appears that the mean curvature κ of the free boundary is the best choice for the algorithm to work effectively in terms of convergence speed and stability, especially in the case of the classical approach.

Notations. In all examples we present below, Σ_k denotes the k^{th} approximation of the optimal free boundary Σ^* , and the quantity $d_H(\Sigma_k, \Sigma)$ denotes the Hausdorff distance between Σ_k and Σ . Also, K denotes the *optimal termination index* when the stopping condition (26) is imposed with some prescribed value for Tol; i.e., $K := \min\{k \in \mathbb{N}_0 : |J(\Sigma_{k+1}) - J(\Sigma_k)| < \text{Tol}\}$. Moreover, for later use, we denote by the index M the last iteration loop of the algorithm before timing out (or equivalently, the maximum number of iterations completed by the algorithm after running it for a specified number of seconds).

5.1 Exterior case

5.1.1 Example 1: Accuracy Tests.

We first test the accuracy of the computed gradient. For this purpose, we consider the exterior Bernoulli problem with

$$\Gamma = C(\mathbf{0}, r), \quad \lambda = \frac{1}{R(\log r - \log R)}, \quad 0 < r < R,$$

where $C(\mathbf{0}, r)$ denotes the circle centered at the origin with radius r . In this case, the only solution is the circle $C(\mathbf{0}, R)$.

We let $r = 0.3$ and $R = 0.5$ (hence, $\Sigma^* = C(\mathbf{0}, 0.5)$), giving us $\lambda = -3.9152$. We take $C(\mathbf{0}, 0.6)$ as the initial guess and compute the optimal shapes using the proposed formulation and the classical Neumann-data tracking approach.

In this example, since the evolution of the free boundary consists of concentric circles, we will often use the term ‘ k^{th} mean radii’ which means the average distance from the origin of the nodes on the exterior boundary of the k^{th} domain Ω_k . Throughout the discussion, this term will be denoted by \bar{R}_k . Furthermore, in all test cases, we take $h_{\max} = 0.02$ and terminate the optimization process after running the algorithm for 300 seconds.

Comparisons of results for different choices of X in (24). In this test case, we present the results of the optimization process when X in the step size formula (24) for t_k is set to either $H^1(\Omega_k)$, $V(\Omega_k)$ or $L^2(\Sigma_k)$ with $\alpha = 0.01$. Figure 1a shows the histories of mean radii of the free boundary obtained using the proposed formulation. Looking at the said figure, it seems that the choice $X = L^2(\Sigma_k)$ provides the fastest convergence to the optimal solution among the three choices. This is primarily due to the fact that when X is set to $L^2(\Sigma_k)$, we have, on the average, larger values for the step size t_k (as expected) than when it is set to either $H^1(\Omega_k)$ or $V(\Omega_k)$ (see Figure 1b). Observe also that the latter two choices almost have the same rate of convergence, which is not surprising since their corresponding norms are actually equivalent. Meanwhile, in Table 1, we tabulate the computational results of the present experiment. The table shows, in particular, the Hausdorff distance between the K^{th}

approximation Σ_K of the free boundary and its exact optimal shape Σ^* , the mean radii \bar{R}_K and its corresponding standard deviation σ_K^{rad} at the K^{th} iterate, where K is the optimal termination index when the tolerance Tol is set to the ones indicated in the first column of the table. Also listed in the table are the values of the cost at the K^{th} iterate and the total computing time to reach convergence when the stopping rule (26) is imposed with Tol values given in the first column of the table. Here we mention that the Tol values listed in the table are actually the values of the tolerance in the stopping rule (26) that were satisfied (omitting the case when 10^{-1} and 10^{-2}) after running the algorithm for 300 seconds (except for the case when $\mathcal{X} = L^2(\Sigma_k)$ where we only present the results up to 10^{-6}). Based on the results, it appears that a reasonable choice for the tolerance Tol when imposing the stopping condition (26) when using the proposed formulation is to take it equal to 10^{-5} . Note also that, for all $\mathcal{X} \in \{H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)\}$, the Hausdorff distance between Σ^* and the computed optimal free boundary Σ_K , with Tol = 10^{-5} in (26), is approximately equal to 0.005.

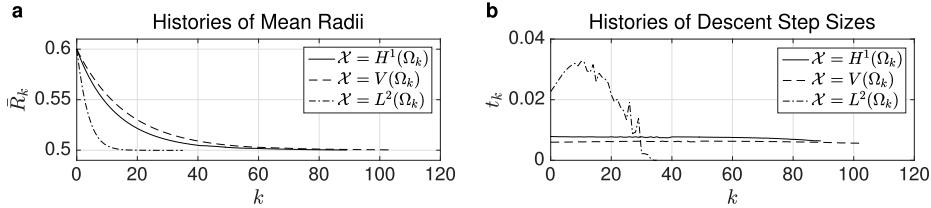


Fig. 1 Histories of mean radii (plot a) and descent step sizes (plot b) when $\mathcal{X} = H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)$ in (24) with $\alpha = 0.01$ using the proposed formulation, running the algorithm for 300 seconds

Tol	$d_H(\Sigma_K, \Sigma^*)$	\bar{R}_K	σ_K^{rad}	$J(\Sigma_K)$	K	CPU time
$\mathcal{X} = H^1(\Omega_k)$						
10^{-3}	0.006660	0.504413	2.33×10^{-5}	0.004413	42	63.89 sec
10^{-4}	0.005371	0.501888	3.74×10^{-5}	0.001888	58	124.21 sec
10^{-5}	0.005004	0.500214	4.05×10^{-5}	0.000214	88	298.37 sec
$\mathcal{X} = V(\Omega_k)$						
10^{-3}	0.008084	0.506344	1.82×10^{-5}	0.038721	46	67.84 sec
10^{-4}	0.005146	0.501220	1.60×10^{-5}	0.001632	78	148.89 sec
10^{-5}	0.005053	0.500766	2.55×10^{-5}	0.000765	89	199.50 sec
$\mathcal{X} = L^2(\Sigma_k)$						
10^{-3}	0.005041	0.500586	3.93×10^{-5}	0.000927	15	23.60 sec
10^{-4}	0.005006	0.500198	4.09×10^{-5}	0.000108	17	29.45 sec
10^{-5}	0.005000	0.500055	2.94×10^{-5}	1.30×10^{-5}	22	49.19 sec
10^{-6}	0.004998	0.500101	3.15×10^{-5}	5.70×10^{-6}	25	74.39 sec

Table 1 Summary of results of the computational experiments when $\mathcal{X} = H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)$ in (24) with $\alpha = 0.01$ using the proposed formulation

On the other hand, the results obtained from using the classical Neumann-data tracking approach are depicted in Figure 2. Figure 2a shows the histories of mean radii \bar{R}_k of the free boundary Σ_k while Figure 2b plots the graph of their corresponding standard deviations. Looking at the latter plot, we observe that the choice $\mathcal{X} = L^2(\Sigma_k)$ gives a very unstable

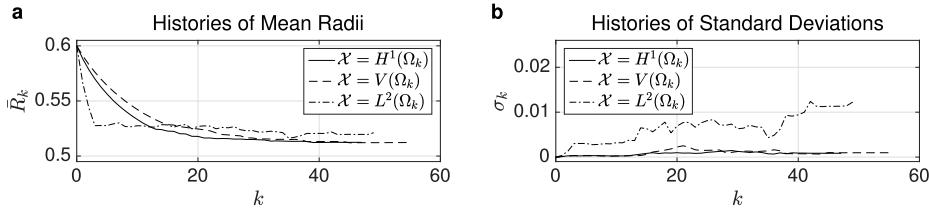


Fig. 2 Histories of mean radii (plot **a**) and their corresponding histories of standard deviations (plot **b**) when $\mathcal{X} = H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)$ in (24) with $\alpha = 0.01$ using the classical Neumann-data tracking approach after 300 seconds of run time

Tol	$d_H(\Sigma_K, \Sigma^*)$	\bar{R}_K	σ_K^{rad}	$J(\Sigma_K)$	K	CPU time
$\mathcal{X} = H^1(\Omega_k)$						
10^{-1}	0.020737	0.517855	0.000972	2.527479	20	22.70 sec
10^{-2}	0.017395	0.514585	0.001157	1.957065	31	38.98 sec
10^{-3}	0.014639	0.512422	0.000855	1.526704	44	195.86 sec
10^{-4}	0.014612	0.512444	0.000848	1.525842	47	288.12 sec
$\mathcal{X} = V(\Omega_k)$						
10^{-1}	0.030236	0.522901	0.002171	4.638160	22	29.40 sec
10^{-2}	0.019315	0.515450	0.001315	2.204293	32	49.77 sec
10^{-3}	0.015384	0.512289	0.000970	1.430624	55	276.75 sec
$\mathcal{X} = L^2(\Sigma_k)$						
10^{-1}	0.038000	0.527682	0.003034	7.934110	5	19.36 sec
10^{-2}	0.044922	0.528277	0.006324	5.996569	15	77.61 sec

Table 2 Summary of results of the computational experiments when $\mathcal{X} = H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)$ in (24) with $\alpha = 0.01$ using the Neumann-data tracking approach

approximation of the free boundary during the optimization process. In fact, we noticed during the optimization process that the exterior boundary Σ becomes very ‘jagged’ after some iterations. This possibly means that the algorithm, when employing the Neumann-data tracking approach, is very sensitive to large deformations, which, on the other hand, suggests that we need to take smaller values for α in order to get more stable approximation of the optimal free boundary. Setting α in (24) to smaller values, however, would then require the algorithm to process additional number of iterations (and therefore demands additional computing times) just in order to attain reasonable approximation of the exact optimal free boundary. Furthermore, even in the case when \mathcal{X} is set to either $H^1(\Omega_k)$ or $V(\Omega_k)$, the histories of mean radii obtained through the Neumann-data tracking approach is less smooth than in the case of when applying the proposed formulation (compare Figure 1a and Figure 2a). Moreover, it is also evident in the two plots shown in Figure 2 that the choices $H^1(\Omega_k)$ and $V(\Omega_k)$ for \mathcal{X} in (24) exhibit almost the same rate of convergence as in the case of using the proposed formulation. It seems, however, that the best choice for the algorithm to work effectively when applying the Neumann-data tracking approach is to take \mathcal{X} as the space $H^1(\Omega_k)$. Meanwhile, the computational results corresponding to the case when using the Neumann-data tracking approach with $\alpha = 0.01$ in (24) are summarized in Table 2. Based on the table, it seems that the appropriate value for the tolerance Tol is 10^{-3} when imposing the stopping rule (26), in case of implementing the Neumann-data tracking approach.

Before we proceed further with our numerical investigations, let us comment and reiterate the most important findings drawn from the results of the computational experiments

presented above. Firstly, it seems that the proposed formulation provides a more stable approximation of the free boundary (in a sense that the domain Ω is less prone to experience rapid oscillating exterior boundary during iterations) than the Neumann-data tracking approach, regardless of the choice of $X \in \{H^1(\Omega_k), V(\Omega_k), L^2(\Sigma_k)\}$. This observation can actually be inferred easily by comparing the order of magnitude of σ_K^{rad} obtained from the two formulations. Secondly, it appears that the proposed formulation exhibits faster convergence behavior than the Neumann-data tracking approach, again irrespective of the choice of X in the step size formula (24). In addition, the former formulation provides better approximation of the analytical solution than the latter approach. Furthermore, it seems that the appropriate choice for Tol when imposing the stopping condition (26) is to take it equal to 10^{-5} when using the proposed formulation and set it to 10^{-3} when applying the Neumann-data tracking approach. In relation to this remark, it appears that the best choice for X in (24) that provides the fastest convergence rate when employing the proposed formulation is the space $L^2(\Sigma_k)$. On the other hand, it seems that the most practical choice for X that provides the most stable and fastest convergence rate when applying the Neumann-data tracking approach is the space $H^1(\Omega_k)$. All these observations can all be inferred easily from the results shown in Table 1 and Table 2, and, of course, from the graphs plotted in Figure 1 and Figure 2.

Examining the order of convergence of the iterative procedure. Next, we numerically examine the convergence behavior of the present iterative scheme. We do this by looking at the sequence of radii of the computed optimal free boundaries obtained from using each of the two formulations. For this purpose, we let $\epsilon_k = |\bar{R}_k - R^*|$ be the error in the k^{th} approximation. Note that, for a ‘good’ numerical procedure, we want the approximate shape solution $\Omega_k := C(\mathbf{0}, \bar{R}_k)$ to be as close as possible to the analytical solution $\Omega^* := C(\mathbf{0}, R^*)$, $R^* = 0.5$. Now, let ρ be the order of convergence of R_k to R^* ; that is, we have that $\lim_{k \rightarrow \infty} \epsilon_{k+1}/\epsilon_k^\rho = \mu$. If we assume that the error progression is exactly of the form $\epsilon_{k+1} = \mu \epsilon_k^\rho$, then we can actually write $\log \epsilon_{k+1} = \rho \log \epsilon_k + \log \mu$. Hence, we can use a best-fit-line approach to find an approximation of ρ , given the sequence of errors ϵ_k . Figure 3 below depicts the order of convergences of the algorithm when using the proposed and the classical Neumann-data tracking approach which correspond to the computational results presented above. It shows, in particular, the order of convergences of the proposed formulation when the stopping condition (26) is imposed with $\text{Tol} = 10^{-5}$ (see Figure 3a and Figure 3b). It also presents the order of convergences of the algorithm when applying the Neumann-data tracking approach which is again terminated via the stopping rule (26) but with $\text{Tol} = 10^{-3}$ (refer to Figure 3c and Figure 3d). In these plots, the *dashed-line* passing through the origin has slope equal to the unity. Meanwhile, the solid-line plot represents the best fit line to the data $\log \epsilon_{k+1} = \rho \log \epsilon_k + \log \mu$ with slope equal to the value indicated in the figure. Clearly, based on these plots, the present algorithm exhibits linear convergence behavior regardless of the formulation used in the optimization procedure.

Effects of increasing the value of the step size parameter α . Let us now look at the effect of increasing the magnitude of the step size t_k in the optimization process by adjusting α to a higher value. Obviously, we could expect that, by increasing the value of α , we could improve the rate of convergence of the algorithm. Such improvement could be expected when employing the proposed formulation in the algorithm (at least for slightly higher values of α for the present case). However, this is not always the case for the other approach since increasing the magnitude of α , in general, would only cause the algorithm to become more unstable. These facts are apparent in the plots shown in Figure 4, and also in Table 3 wherein the results of the optimization process obtained through the proposed and the classical Neumann-data tracking approach with $X = H^1(\Omega)$ and $\alpha \in \{0.02, 0.03, 0.04\}$ are summarized. In the table, the notation \bar{t} represents the computed mean step size for the en-

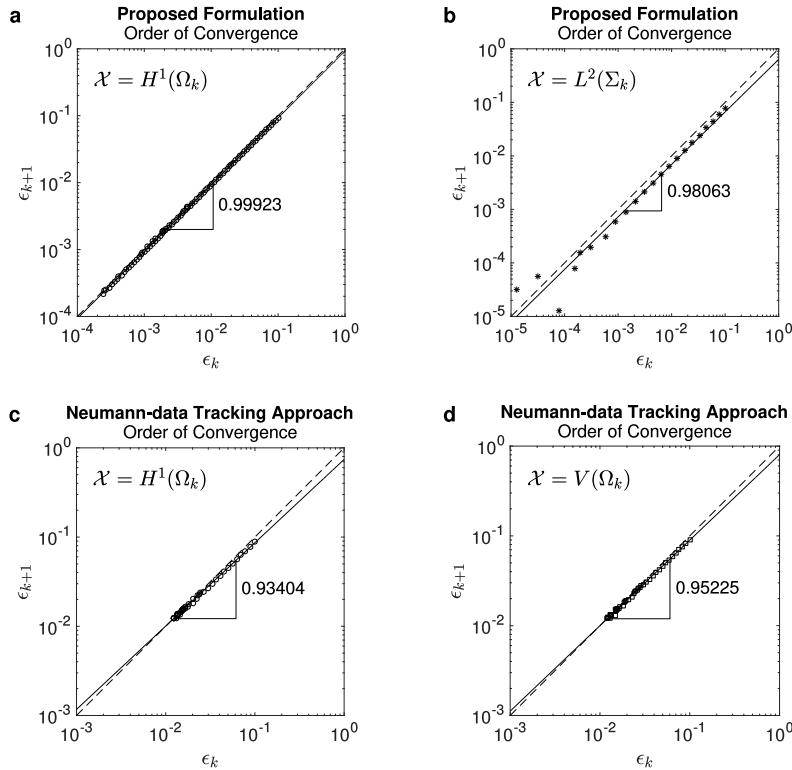


Fig. 3 Order of convergences of the algorithm when applying the proposed formulation (plots **a** and **b**) and when using the Neumann-data tracking approach (plots **c** and **d**) with $\alpha = 0.01$, employing the stopping condition (26) with $Tol = 10^{-5}$ and 10^{-3} for the first and second approach, respectively

tire optimization process when the stopping rule (26) is utilized (i.e., $\bar{t} = \sum_{k=0}^K t_k / K$). The notation σ^{step} , on the other hand, denotes the standard variation of the computed step sizes, also for the entire iteration process. Meanwhile, the last column in Table 3 indicates the coefficient of variation (C.V.) with respect to the step size t_k , i.e., $\text{C.V.} = \sigma^{\text{step}} / \bar{t}$. In relation to this, we note that having a coefficient of variation for the step size that is larger than the unity is an indication that the step size varies greatly from each iterate. This means, possibly, that the algorithm is very much less stable, and based on our experience, large fluctuations in step size could cause the algorithm to crash during iterations. These results further support our claim that the proposed formulation provides more stable approximation of the exact optimal solution than the classical Neumann-data tracking approach. Moreover, it is clear from the table that the former approach is more accurate than the latter one. Lastly, notice from the table that the number of iterations (and hence, the computing times) required to reach convergence when imposing the stopping condition (26) is significantly less for the case of the proposed formulation than in the case of using the Neumann-data tracking approach.

In the rest of the examples below, we utilize the main findings drawn above. More precisely, we take $\mathcal{X} = L^2(\Sigma_k)$ in (24) when the proposed formulation is being applied in the algorithm and, on the other hand, set $\mathcal{X} = H^1(\Omega_k)$ when the Neumann-data tracking approach is used in the optimization procedure.

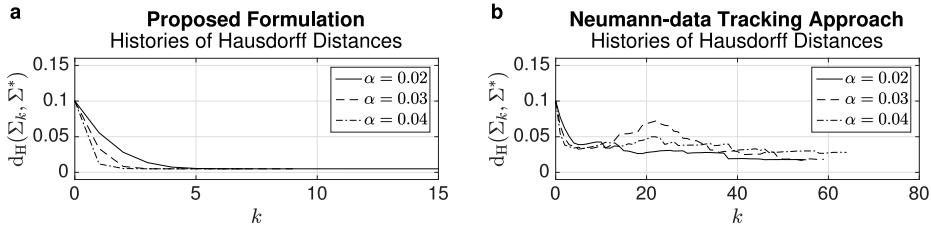


Fig. 4 Histories of Hausdorff distances using the proposed formulation with $\mathcal{X} = L^2(\Sigma_k)$ (plot a) and when applying the Neumann-data tracking approach with $\mathcal{X} = H^1(\Omega_k)$ for some values of α

α	$d_H(\Sigma_K, \Sigma^*)$	\bar{R}_K	σ_K^{rad}	$J(\Sigma_K)$	K	time	\bar{t}	σ^{step}	C.V.
Proposed approach with $\mathcal{X} = L^2(\Sigma_k)$ in (24) and terminated using (26) with $\text{Tol} = 10^{-5}$									
0.02	0.005002	0.499914	0.000037	4.81×10^{-6}	10	52 s	0.0375	0.0235	< 1
0.03	0.005002	0.499688	0.000062	3.09×10^{-7}	7	21 s	0.0524	0.0458	< 1
0.04	0.005014	0.499620	0.000074	1.49×10^{-9}	7	24 s	0.0926	0.0650	< 1
Neumann-data tracking approach with $\mathcal{X} = H^1(\Sigma_k)$ in (24) and stopped using (26) with $\text{Tol} = 10^{-3}$									
0.02	0.017956	0.513706	0.001123	1.697592	51	183 s	0.0009	0.0012	> 1
0.03	0.018772	0.513442	0.001803	1.721732	52	276 s	0.0012	0.0014	> 1
0.04	0.029857	0.515549	0.005606	1.826101	54	116 s	0.0010	0.0016	> 1

Table 3 Summary of results of the computational experiments corresponding to Figure 4

5.1.2 Example 2: A T-shaped fixed boundary

Next, we consider $\Gamma = \partial S$ as the boundary of the T-shape

$$S := ((-3/8, 3/8) \times (-1/4, 0)) \cup ((-1/8, 1/8) \times [0, 1/4]).$$

The optimal domain for $\lambda = -1, -2, \dots, -10$. First, we compute the optimal domain for all integers $\lambda = -1, -2, \dots, -10$ using the proposed formulation. We choose the unit circle as the initial guess and let $h_{\max} = 0.025$ for all cases. Furthermore, we terminate the iteration process after running the algorithm for 60 seconds. The resulting exterior boundaries are shown in Figure 5, where the outermost boundary corresponds to $\lambda = -1$ and the innermost boundary to $\lambda = -10$, and the shaded area represents the region bounded by the fixed boundary. Meanwhile, the results of the present computational experiments are summarized in Table 4 when the stopping rule (26) with $\text{Tol} = 10^{-5}$ is used. The table shows in particular the computed cost value at the K^{th} iterate and the total computing time to reach convergence for each values of λ . Also indicated in the table are the total number of iterations completed by the algorithm before timing out. The values shown in Table 4 were all obtained with the step size parameter α set to 0.10 except for the case when $\lambda = -3$ where we slightly adjusted α to 0.11 to reach convergence under the stopping condition (26) with $\text{Tol} = 10^{-5}$ and within 60 seconds. On the other hand, we remark that the optimal free boundaries obtained when the stopping rule (26) is employed are actually indistinguishable from the ones shown in Figure 5 (see Figure 6 for a direct comparison of the free boundaries Σ_K and Σ_M when $\lambda = -1, -8$). The evolution of the free boundary when $\lambda = -10$ is shown in Figure 7a.

Comparison of results obtained from the two formulations. Next, we compare the optimal free boundaries obtained from the two formulations for the present test case. We focus particularly on the case when $\lambda = -10$. So, we repeat the optimization process for the case

$\lambda = -10$, but now using the Neumann-data tracking approach. Also, this time, we run the algorithm for 120 seconds and again take $\alpha = 0.10$. In contrast to the evolution of the free boundary shown in Figure 7a, we notice several oscillations appearing on Σ_k , at some iterations, when using the Neumann-data tracking approach. These unwanted irregularities on the free boundary are actually discernible from the evolution of Σ_k obtained through the said approach shown in Figure 7b. We emphasize that such phenomenon actually indicates that the algorithm is unstable for large deformations (which has already been observed in Example 5.1.1). These oscillations appearing on the free boundary during iterations can actually be avoided by taking smaller values for the step size; that is, by reducing the magnitude of α , in expense, of course, of processing additional number of iterations to attain good enough approximation of the optimal free boundary. In relation to this, the evolution of the free boundary using the classical approach under the same setup, but now with $\alpha = 0.01$, is shown in Figure 7c. Observe that, with the new value of α , we now have a smooth evolution of the free boundary (but smaller gaps between each consecutive shape deformations). Meanwhile, a direct comparison of the computed optimal free boundary Σ_M obtained from the two formulations (with $\alpha = 0.10$) are shown in Figure 7d. For the proposed formulation, the final cost value is $J(\Sigma_{23}) = 2.99 \times 10^{-8}$ and for the Neumann-data tracking approach, we obtain the value $J(\Sigma_{80}) = 4.10$ at the final iteration. We mention here that we have not actually satisfied the stopping condition (26) with $Tol = 10^{-3}$ when using the Neumann-data tracking approach, after running the algorithm for 120 seconds. Nevertheless, we are able to satisfy (26) for $Tol = 10^{-2}$ after $K = 42$ iterations (which was completed after 47.93 seconds of run time) with cost value $J(\Sigma_{42}) = 5.06$. Taking $Tol = 10^{-2}$ as the tolerance value when imposing the stopping rule (26), however, seems reasonable since the cost actually decreases very slow after reaching 39 iterations, as evident in Figure 8.

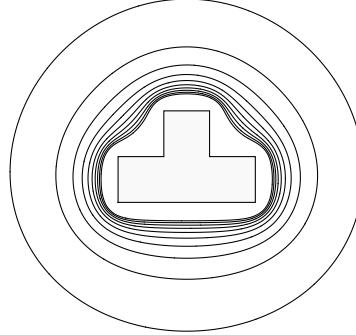
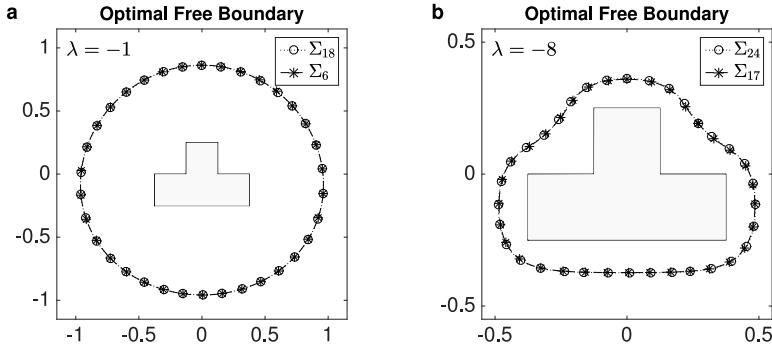


Fig. 5 The optimal free (or exterior) boundaries Σ^* using the proposed formulation

Remark 1 We remark here that we actually took slightly larger edge size for the mesh (in fact, we set $h_{\max} = 0.03$) during mesh adaptation in performing the optimization process with the Neumann-data tracking approach. The main reason for this different setup is that it is actually difficult to obtain stable approximation of the optimal free boundary when using the said approach with finer mesh during mesh adaptation. In fact, the algorithm crashes after a certain number of iterations when a smaller value for h_{\max} is used during mesh adaptation.

Below we provide a few more examples illustrating further the robustness of the proposed formulation in solving the exterior Bernoulli free boundary problem.

λ	$J(\Sigma_K)$	K	CPU time	M
-1	3.68×10^{-5}	6	11.98 sec	18
-2	7.84×10^{-6}	13	22.21 sec	17
-3	5.37×10^{-5}	10	16.55 sec	16
-4	6.27×10^{-6}	14	27.72 sec	16
-5	1.51×10^{-5}	16	25.56 sec	20
-6	4.59×10^{-7}	15	31.99 sec	17
-7	9.52×10^{-6}	21	40.14 sec	24
-8	2.03×10^{-6}	17	41.61 sec	24
-9	1.94×10^{-6}	22	42.09 sec	24
-10	1.22×10^{-7}	21	55.82 sec	23

Table 4 Summary of computational results corresponding to the optimal free boundaries shown in Figure 5**Fig. 6** Optimal free (or exterior) boundaries obtained through the proposed formulation for Example 5.1.2 (the case where $\Gamma = \partial S$) when $\lambda = -1$ (plot a) and when $\lambda = -8$ (plot b)

5.1.3 Example 3: An L-shaped fixed (interior) boundary

In this test case, we consider the boundary $\Gamma = \partial S$ of the L-shaped domain $S = (-0.25, 0.25)^2 \setminus [0, 0.25]^2$ and let $\lambda = -10$. For the initial guess, we select the circle $C(0, 0.6)$ and take $\alpha = 0.10$ for both approaches. Also, we run the algorithm for 300 seconds and we let $h_{\max} = 0.025$ when using the proposed formulation and take $h_{\max} = 0.03$ when applying the Neumann-data tracking approach. In case of using the proposed formulation, the algorithm completed 16 iterations (which was reached after 23.49 seconds of run time) with the final cost value $J(\Sigma_{16}) = 4.68 \times 10^{-11}$. The stopping rule (26) with $Tol = 10^{-5}$ is satisfied after 15 iterations which was completed after 19.77 seconds and the computed cost is $J(\Sigma_{15}) = 2.17 \times 10^{-7}$. The evolution of the free boundary obtained through the proposed formulation is shown in Figure 9a. On the other hand, when the Neumann-data tracking approach is applied, the optimization procedure completely processed 158 iterations with the final cost value $J(\Sigma_{158}) = 0.88$. Again, during the iteration process, we notice several oscillations on the free boundary, and these oscillations are noticeable from the evolution of the free boundary obtained through Neumann-data tracking approach depicted in Figure 10b. A direct comparison of the optimal free boundaries Σ_{16} and Σ_{158} obtained through the proposed formulation and from the classical Neumann-data tracking approach, respectively, are shown in Figure 10c. Observe that the two computed optimal free boundaries are almost

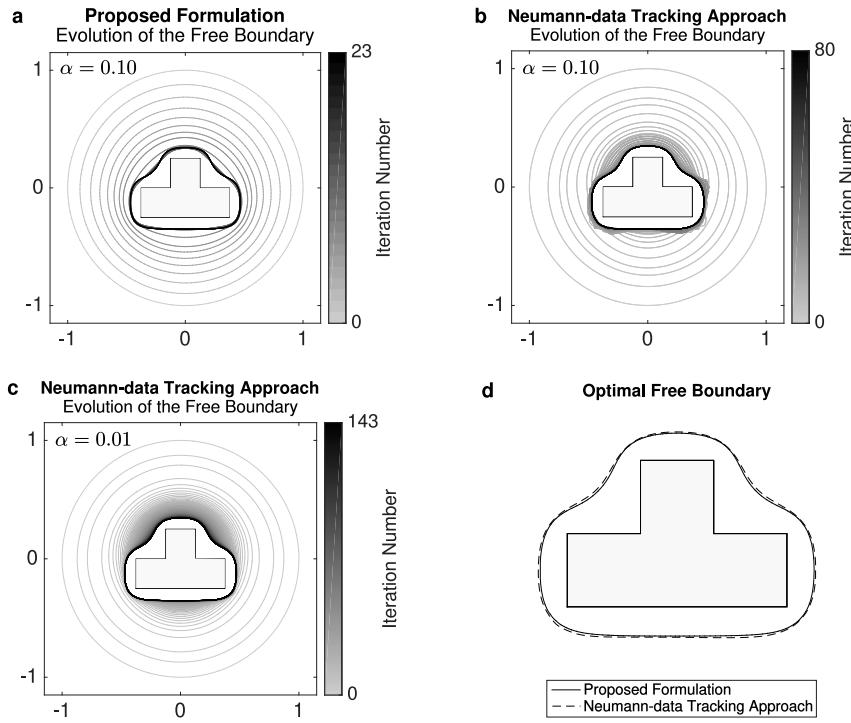


Fig. 7 Evolutions of the free boundary for the case $\lambda = -10$ when the proposed formulation is employed with $\alpha = 0.10$ (plot **a**) and when the Neumann-data tracking approach is applied with $\alpha = 0.10, 0.01$ (plots **b** and **c**, respectively); **d**: direct comparison between the optimal free boundaries obtained through the proposed and the classical Neumann-data tracking approach when $\alpha = 0.10$

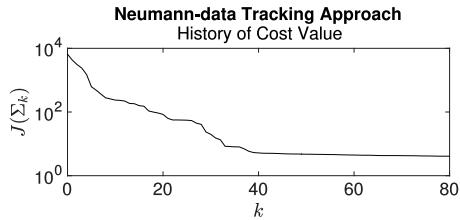


Fig. 8 Convergence history of the cost function for Example 5.1.2 when $\lambda = -10$, applying the Neumann-data tracking approach with $\alpha = 0.10$

indistinguishable from each other. However, it seems that Σ_{158} is slightly larger compared to Σ_{16} . The Hausdorff distance between the two is computed to be of order 10^{-2} . Meanwhile, for the classical approach, the stopping rule (26) with $Tol = 10^{-2}$ (respectively, 10^{-3}) is met after 49 (respectively, 139) iterations which was attained after 48.77 seconds (respectively, 244.66 seconds) of run time. Similar to the case of the previous example, setting the tolerance value in (26) to $Tol = 10^{-2}$ seems reasonable if the stopping rule is applied because the cost actually decreases slowly after reaching 49 iterations (see upper plot in Figure 9d). This observation is also apparent from the history of Hausdorff distances between the k^{th} and the 139th approximations of the optimal free boundary depicted in the lower plot in Figure 9d.

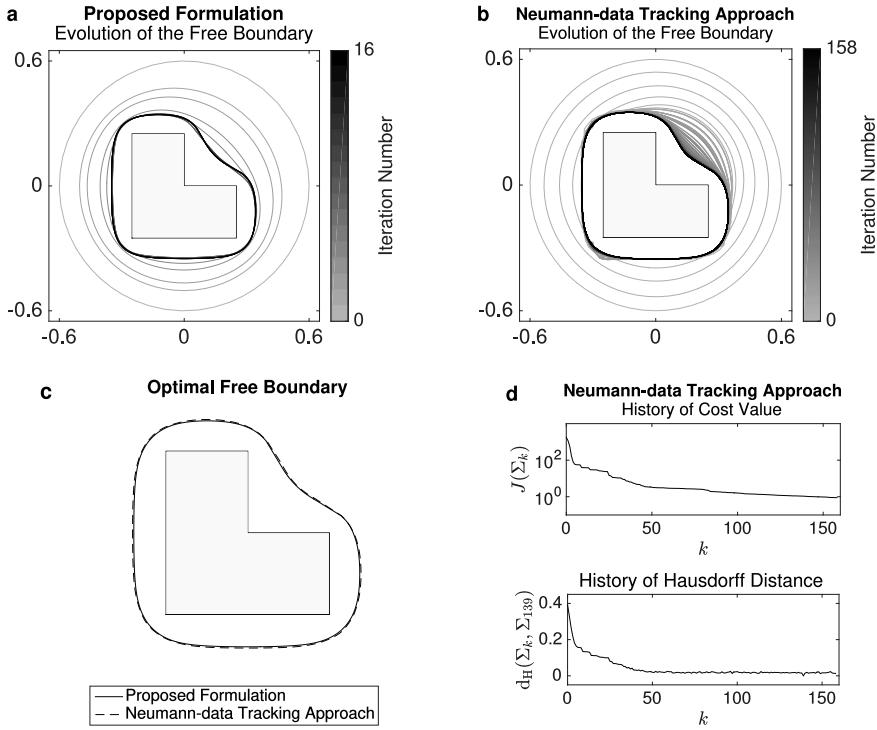


Fig. 9 Evolutions of the free boundary for Example 5.1.3 when the proposed formulation is employed (plot a) and when the Neumann-data tracking approach is applied (plot b) where $\alpha = 0.10$ in both cases; c: direct comparison between the optimal free boundaries obtained through the proposed and the classical Neumann-data tracking approach when $\alpha = 0.10$; d: histories of cost values (upper plot) and Hausdorff distances (lower plot) obtained through the Neumann-data tracking approach corresponding to plot b

5.1.4 Example 4: A fixed boundary with two disjoint components

For the last example under the exterior case, we take $\lambda = -1.5$ and define the fixed boundary as the union of two disjoint kite-shaped figures given by the following parametrization

$$\begin{aligned}\Gamma_1 &= \{(1 + 0.7 \cos \theta - 0.4 \cos 2\theta, \sin \theta)^\top, 0 \leq \theta \leq 2\pi\}, \\ \Gamma_2 &= \{(-2 + \cos \theta + 0.4 \cos 2\theta, 0.5 + 0.7 \sin \theta)^\top, 0 \leq \theta \leq 2\pi\}.\end{aligned}$$

For the initial guess, we choose the circle $C(\mathbf{0}, 5)$. In this case, the solution is known to be connected, hence the present scheme is suitable for numerically solving the problem (cf. [31]). Here, in all situations, the algorithm is ran for 600 seconds. Also, we use finer mesh during mesh adaptation when employing the proposed formulation than when applying the Neumann-data tracking approach. In particular, we use $h_{\max} = 0.05$ for the former approach and set $h_{\max} = 0.10$ for the latter method. We choose coarser mesh when using the Neumann-data tracking approach for the same reason stated in Remark 1.

The evolution of the free boundary obtained through the application of the proposed formulation is depicted in Figure 10 when $\alpha = 0.10, 0.25$ and 0.50 (see plots a, b and c, respectively). Clearly, as α increases in magnitude, the convergence speed also increases (of course, this is only true up to some value of α as in the results shown in Table 3). In the same

figure, particularly, in plots b, d and f, the evolution of the free boundary obtained using the Neumann-data tracking approach when $\alpha = 0.001, 0.010, 0.100$ are shown. Notice that, even at small step sizes, the free boundary is prone to oscillations as evident (although not too visible) in Figure 10b. Also, it is apparent from Figures 10d and f that increasing α , in case of using the Neumann-data tracking approach, only worsen the oscillations appearing on the free boundary during the optimization process. The numerical results of the present computational experiments are summarized in Table 5. The table shows, in particular, the computed cost value $J(\Sigma_K)$ at the optimal termination number K when the stopping condition (26) is utilized with tolerance value $Tol = 10^{-5}$ for the proposed formulation and $Tol = 10^{-2}$ for the Neumann-data tracking approach (the reason behind these choices of Tol values will be issued later). Also, listed in the table are the corresponding final cost values $J(\Sigma_M)$ for each of the methods applied (and for each values of α used in the experiment). Surprisingly, the computing time to reach convergence (imposing the stopping rule (26)) when using the proposed formulation with $\alpha = 0.25$ is almost the same with the case when α is set to 0.50. Meanwhile, the corresponding histories of cost values of the free boundaries shown in Figure 10 are plotted in Figure 11a. Observe that, in case of the Neumann-data tracking approach, the values 0.01 and 0.10 for α exhibits comparable convergence speed. Based on this, it seems that the rate of convergence of the optimization process when applying the Neumann-data tracking approach could not be further improved even when α is increased in magnitude. On the other hand, Figures 11b and c respectively plots the histories of Hausdorff distances between Σ_k and the final computed free boundary Σ_M obtained through the proposed formulation, for each $\alpha = 0.10, 0.25, 0.50$, and via the Neumann-data tracking approach, for each $\alpha = 0.001, 0.010, 0.100$. In these plots, the abbreviation ‘H.D.’ appearing on the vertical axes means the term ‘Hausdorff Distance’. In all situations, including the cross comparisons between the final optimal free boundaries obtained from using each of the values of α and the comparisons between the results obtained from each methods, the computed Hausdorff distances are all of order 10^{-2} (or lower). We emphasize that the said order of magnitude of the computed Hausdorff distances is reasonable since we used coarser mesh during the optimization process. Furthermore, the graphs depicted in Figures 11b and c show that the stopping rule (26) can indeed be effectively used to terminate the iteration process by taking the tolerance value $Tol = 10^{-5}$ when using the proposed formulation and setting it to $Tol = 10^{-2}$ when the Neumann-data tracking approach is being applied. Finally, a direct comparison between the computed optimal free boundaries Σ_{32} and Σ_{114} obtained through the proposed formulation (with $\alpha = 0.50$) and the classical Neumann-data tracking approach (with $\alpha = 0.10$), respectively, are shown in Figure 11d. As in Example 5.1.2, it seems that the optimal free boundary due to the Neumann-data tracking approach is slightly larger compared to the one obtained through the proposed formulation.

5.2 Interior case

We now provide some numerical examples for the interior case. This time we take $X = L^2(\Omega_k)$ and let $\alpha = 0.99$, for simplicity, in the step size formula (24) for both the proposed and the classical Neumann-data tracking approach. Also, we run the algorithm for 60 and 600 seconds in the first (Example 5.2.1) and second test case (Example 5.2.2), respectively.

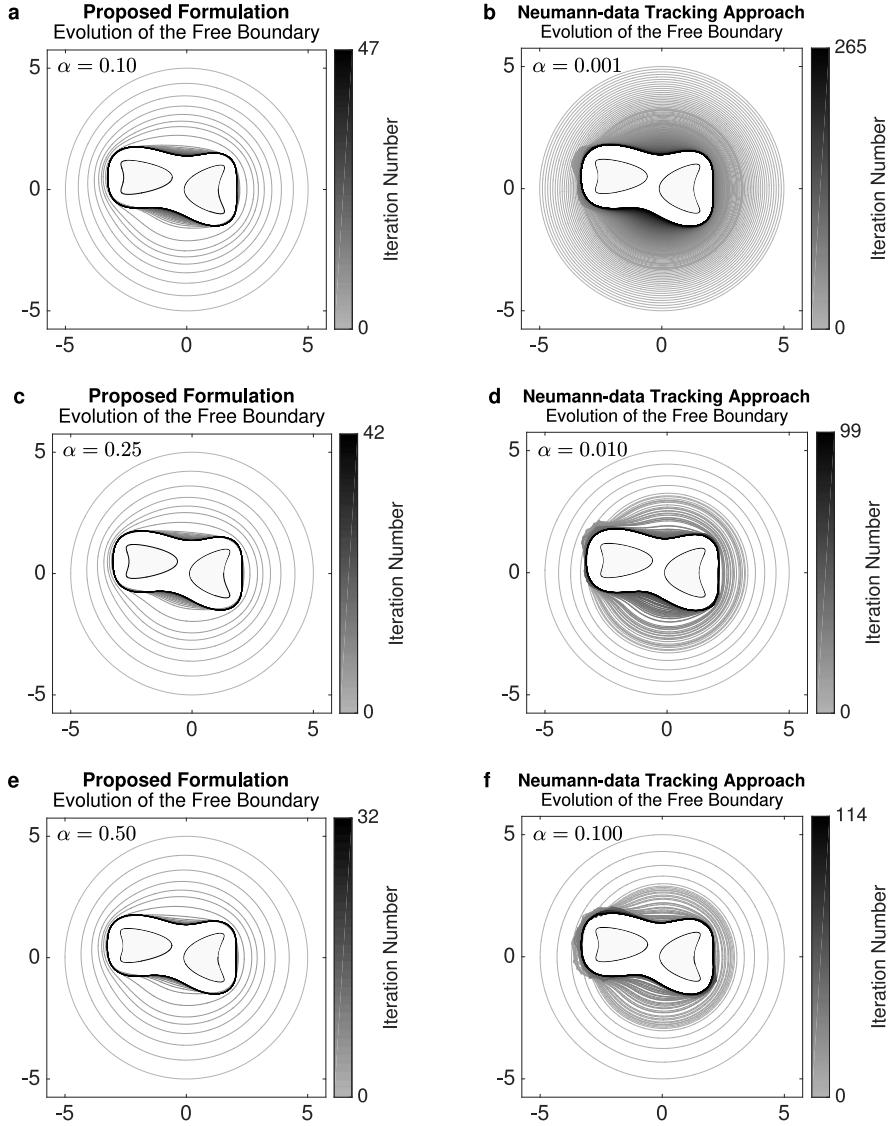


Fig. 10 Evolutions of the free boundary corresponding to Example 5.1.4 when the proposed formulation is employed with $\alpha = 0.10, 0.25, 0.50$ (plots a, c and e, respectively) and when the Neumann-data tracking approach is applied with $\alpha = 0.001, 0.010, 0.100$ (plots b, d and f, respectively)

5.2.1 Example 5: Accuracy Tests.

For the first test case, we again check the accuracy of the computed gradient. To this end, we consider the interior Bernoulli problem with

$$\Gamma = C(\mathbf{0}, R), \quad \lambda = \frac{1}{r(\log R - \log r)}, \quad R/e < r < R.$$

α	$J(\Sigma_K)$	K	CPU time	$J(\Sigma_M)$	M
Proposed formulation; Tol = 10^{-5}					
0.10	0.000762	38	146.68 sec	0.000473	47
0.25	0.000164	27	55.30 sec	1.26×10^{-6}	42
0.50	2.20×10^{-5}	22	55.28 sec	1.30×10^{-8}	32
Neumann-data tracking approach; Tol = 10^{-2}					
0.001	9.25	210	363.67 sec	5.73	265
0.010	8.02	71	86.95 sec	7.43	99
0.100	5.69	68	187.94 sec	4.60	114

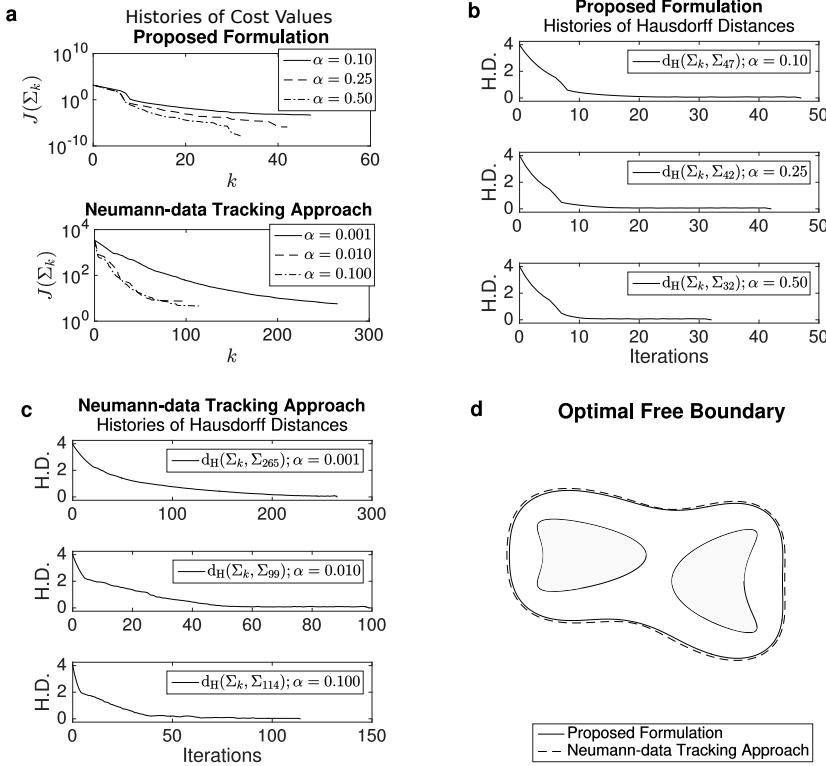
Table 5 Summary of computational results corresponding to Example 5.1.4

Fig. 11 **a:** Convergence histories of the cost function for both the proposed and classical formulations with different values of α ; the histories of Hausdorff distances between Σ_k and the final computed free boundary Σ_M obtained through the proposed formulation, for each $\alpha = 0.10, 0.25, 0.50$ (plot **b**), and via the Neumann-data tracking approach, for each $\alpha = 0.001, 0.010, 0.100$ (plot **c**); **d:** direct comparison between the optimal free boundaries obtained through the proposed and the classical Neumann-data tracking approach

For this case, the interior Bernoulli problem admits two possible solutions; namely, the *elliptic* solution which is the circle $C(\mathbf{0}, r)$, and the *hyperbolic* solution given by the circle $C(\mathbf{0}, r_h)$, where r_h is the unique real number such that

$$0 < r_h < R/e, \quad \frac{1}{r_h(\log R - \log r_h)} = \lambda.$$

Convergence to the elliptic or hyperbolic solution depends on the initial guess. In our test, we are interested only in the elliptic solution. We take $R = 0.9$ and $r = 0.5$, so $\lambda = 3.4026$ (and again, obviously, $\Sigma^* = C(\mathbf{0}, 0.5)$). We choose $C(\mathbf{0}, 0.6)$ as the initial guess. The histories of the mean radii and Hausdorff distances obtained through the application of the proposed formulation are plotted in Figures 12a and b, respectively, for some mesh sizes h_{\max} used during mesh adaptation. Observe from the said plots that the rate of convergence of the mean radii and of the Hausdorff distances slows down after four iterations. If we imposed the stopping condition (26), with $\text{Tol} = 10^{-4}$, the algorithm actually terminates after six iterations, irrespective of the magnitude of maximum edge size of the mesh h_{\max} used during mesh adaptation. Table 6 summarizes the results toward the elliptic solution when the stopping rule (26) with $\text{Tol} = 10^{-4}$ is utilized to terminate the algorithm. It is clear from the table that the accuracy of the computed optimal free boundary is improved as the magnitude of h_{\max} is reduced. Also, based on the computed value for the standard deviation σ_K^{rad} shown in the said table, we can actually say that the proposed method produces a very stable approximation of the optimal solution, in a sense that every domain Ω_k , $k = 1, 2, \dots, M$, has an exterior boundary Σ_k with no rapid oscillation. We have also ran the algorithm using the Neumann-data tracking approach. However, the algorithm was only able to process one complete iteration (regardless of the magnitude of h_{\max}) and the computed free boundary has mean radius $\bar{R}_1 = 0.5127$, Hausdorff distance of $d_H(\Sigma_1, \Sigma^*) = 0.0207$ from the exact optimal shape Σ^* and final cost value $J(\Sigma_1) = 0.0046$ when h_{\max} is set to 1/160. It seems that the formula (24) produces a very small magnitude for t_1 which is already of order 10^{-3} , and apparently, this step size is not large enough to produce a variation of the current domain Ω_1 that would decrease the magnitude of the cost at the next iteration. In addition, we mention that the computed shape at the first iterate actually has some irregularities appearing on its exterior boundary. In fact, the computed standard deviation σ_1^{rad} is equal to 0.0029 which is one order higher compare to the order of magnitude of σ_K^{rad} 's listed in Table 6.

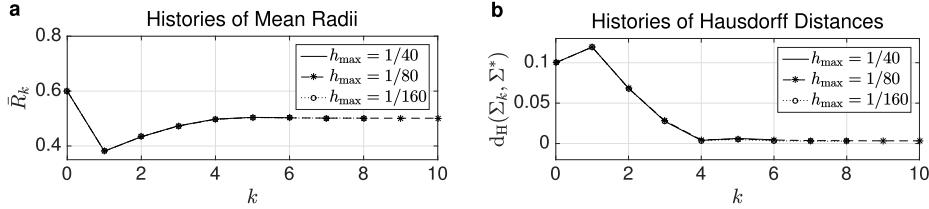


Fig. 12 Histories of mean radii (plot a) and Hausdorff distances (plot b) obtained through the application of the proposed formulation after running the algorithm for 60 seconds

h_{\max}	$d_H(\Sigma_K, \Sigma^*)$	\bar{R}_K	σ_K^{rad}	$J(\Sigma_K)$	K	time	\bar{t}	σ^{step}	C.V.
1/40	0.004602	0.502944	0.000292	7.27×10^{-6}	6	12.83 s	0.6327	0.3545	< 1
1/80	0.004341	0.501177	0.000205	7.46×10^{-6}	6	17.63 s	0.5907	0.2780	< 1
1/160	0.003328	0.501606	0.000193	1.91×10^{-6}	6	17.37 s	0.6119	0.3010	< 1

Table 6 Summary of computational results corresponding to Example 5.2.1 using the proposed formulation terminated with the stopping rule (26) with $\text{Tol} = 10^{-4}$

5.2.2 Example 6: An L-shaped fixed (exterior) boundary

For the second test case, we consider the boundary $\Gamma = \partial S$ of the L-shaped domain $S = (-0.5, 0.5)^2 \setminus [0.1, 0.5]^2$ and take $\lambda = 14$. We choose $C((-0.15, -0.15), 0.25)$ as the initial guess. The result of the computational experiments are summarized in Table 7. Clearly, based from the table, the proposed formulation converges significantly faster to the optimal solution than the classical Neumann-data tracking approach. The evolution of the free boundary using the proposed and the classical Neumann-data tracking approach are respectively depicted in Figures 13a and b (where the shaded region represents the final computed domain Ω_M). Observe from the latter figure that the there are some irregularities appearing on the shape of the free boundary at several iterations. Meanwhile, a direct comparison between the free boundaries Σ_{17} and Σ_{104} , respectively obtained through the proposed formulation and the Neumann-data tracking approach, are shown in Figure 13c. These shapes have the corresponding cost values $J(\Sigma_{17}) = 1.55 \times 10^{-7}$ and $J(\Sigma_{104}) = 0.33$. Also, the computed Hausdorff distance between Σ_{17} and Σ_{104} are found to be equal to 0.02. Lastly, in Figure 13d, we plot the histories of cost values and Hausdorff distances $d_H(\Sigma_k, \Sigma_{104})$ obtained through the Neumann-data tracking approach. Notice that the value of $d_H(\Sigma_k, \Sigma_{104})$ fluctuates at a certain number after 44 iterations. So, based from Table 7, we can actually terminate the algorithm using (26) with $Tol = 10^{-2}$. On the other hand, the value $Tol = 10^{-5}$ seems a reasonable choice for the tolerance value when imposing the stopping rule (26).

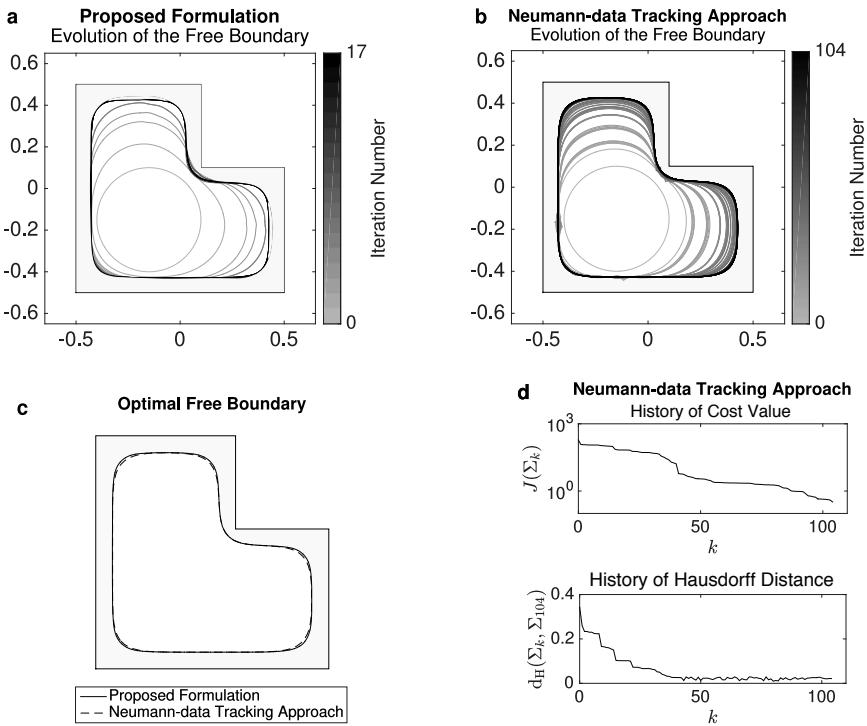


Fig. 13 Evolutions of the free boundary for Example 5.2.2 when the proposed formulation is employed (plot a) and when the Neumann-data tracking approach is applied (plot b); c: direct comparison between the optimal free boundaries obtained through the proposed and the classical Neumann-data tracking approach; d: convergence history of the function (upper plot) and history of Hausdorff distances (lower plot) obtained through the Neumann-data tracking approach corresponding to plot b

Tol	$J(\Sigma_K)$	K	$d_H(\Sigma_K, \Sigma_M)$	CPU time
Proposed formulation				
10^{-3}	0.037359	7	0.017952	35.73 sec
10^{-4}	7.83×10^{-5}	11	0.004287	71.17 sec
10^{-5}	7.12×10^{-5}	12	0.004174	73.77 sec
Neumann-data tracking approach				
10^{-1}	112.97	7	0.223435	37.38 sec
10^{-2}	2.41	57	0.019425	194.46 sec
10^{-3}	2.23	70	0.025111	430.145 sec

Table 7 Summary of computational results corresponding to Example 5.2.2

6 Conclusion

We presented a new shape optimization formulation of the Bernoulli problem by tracking the Neumann data in a least-squares sense. The novelty of the present investigation was the use of a mixed Dirichlet-Robin problem as the state equation which provides more regularity to the associated adjoint state. We numerically solved the optimization problem through an iterative procedure by steepest descent using the knowledge of the shape gradient of the cost combined with a Lagrangian-like method. Numerical tests revealed positive implications of the new formulation on the accuracy, convergence speed and stability of the algorithm. This led us to conclude that our proposed approach is more robust compared to that of the classical shape optimization setting by tracking the Neumann data in numerically solving the well-known Bernoulli free boundary problem.

Acknowledgements The authors wish to thank the anonymous referee for carefully handling and examining the previous version of the manuscript. His/her valuable comments and suggestions greatly improved the presentation and quality of the paper. The first author greatly acknowledges the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for scholarship support during his PhD program.

Appendix: The theorem of Correa and Seeger

We first introduce some notations. Consider a functional $G : [0, \varepsilon] \times X \times Y \rightarrow \mathbb{R}$, for some $\varepsilon > 0$ and the topological spaces X and Y . For each $t \in [0, \varepsilon]$, define

$$g(t) = \inf_{x \in X} \sup_{y \in Y} G(t, x, y) \quad \text{and} \quad h(t) = \sup_{y \in Y} \inf_{x \in X} G(t, x, y),$$

and the associated sets

$$X(t) = \left\{ \hat{x} \in X : \sup_{y \in Y} G(t, \hat{x}, y) = g(t) \right\} \quad \text{and} \quad Y(t) = \left\{ \hat{y} \in Y : \inf_{x \in X} G(t, x, \hat{y}) = h(t) \right\}.$$

Given the above definitions, we introduce the *set of saddle points* $S(t) = \{(\hat{x}, \hat{y}) \in X \times Y : g(t) = G(t, \hat{x}, \hat{y}) = h(t)\}$, which may be empty. In general, the inequality $h(t) \leq g(t)$ holds and when $h(t) = g(t)$, we exactly have $S(t) = X(t) \times Y(t)$. Here, we are particularly interested on the situation when G admits saddle points for all $t \in [0, \varepsilon]$.

Now, we quote an improved version [21, Theorem 5.1, pp. 556–559] of the theorem of Correa and Seeger. The result provides realistic conditions under which the existence of the limit

$$dg(0) = \lim_{t \searrow 0} \frac{g(t) - g(0)}{t}$$

is guaranteed.

Theorem 1 ([25]) Let X , Y , G and ε be given as previously. Assume that the following assumptions hold:

- (H1) for all $t \in [0, \varepsilon]$, the set $S(t)$ is non-empty;
- (H2) the partial derivative $\partial_t G(t, x, y)$ exists for all $(t, x, y) \in [0, \varepsilon] \times X \times Y$;
- (H3) for any sequence $\{t_n\}_{n \in \mathbb{N}}$, with $t_n \rightarrow 0$, there exists a subsequence $\{t_{n_k}\}_{k \in \mathbb{N}}$ and $x^0 \in X(0)$, $x_{n_k} \in X(t_{n_k})$ such that for all $y \in Y(0)$, $\liminf_{t \searrow 0} \partial_t G(t, x_{n_k}, y) \geq \partial_t G(0, x^0, y)$;
- (H4) for any sequence $\{t_n\}_{n \in \mathbb{N}}$, with $t_n \rightarrow 0$, there exists a subsequence $\{t_{n_k}\}_{k \in \mathbb{N}}$ and $y^0 \in Y(0)$, $y_{n_k} \in Y(t_{n_k})$ such that for all $x \in X(0)$, $\limsup_{t \searrow 0} \partial_t G(t, x, y_{n_k}) \leq \partial_t G(0, x, y^0)$.

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

$$\frac{dg}{dt}(0) = \partial_t G(0, x^0, y^0).$$

References

1. Lacey AA, Shillor M (1987) Electrochemical and electro-discharge machining with a threshold current. *IMA J Appl Math* 39:131–142
2. Friedman A (1984) Free boundary problem in fluid dynamics. *Astérisque*. Soc Math France 118:55–86
3. Friedman A (2000) Free boundary problems in science and technology, *Notices of the AMS* 47:854–861
4. Acker A (1981) An extremal problem involving distributed resistance. *SIAM J Math Anal* 12:169–172
5. Acker A (1977) Heat flow in equalities with applications to heat flow optimization problems. *SIAM J Math Anal* 8:604–618
6. Crank J (1984) *Free and Moving Boundary Problems*. Oxford University Press, New York
7. Fasano A (1992) Some free boundary problems with industrial applications. In: Delfour M (ed) *Shape optimization and free boundaries*, vol 380. Kluwer, Dordrecht
8. Tiihonen T (1997) Shape optimization and trial methods for free boundary problems. *ESAIM Math Model Num* 31(7):805–825
9. Haslinger J, Kozubek T, Kunisch K, Peichl GH (2003) Shape optimization and fictitious domain approach for solving free-boundary value problems of Bernoulli type. *Comput Optim Appl* 26:231–251
10. Ito K, Kunisch K, Peichl GH (2006) Variational approach to shape derivative for a class of Bernoulli problem. *J Math Anal Appl* 314:126–149
11. Haslinger J, Ito K, Kozubek T, Kunisch K, Peichl GH (2003) On the shape derivative for problems of Bernoulli type. *Interfaces Free Bound* 11:317–330
12. Eppler K, Harbrecht H (2009) Tracking Neumann data for stationary free boundary problems. *SIAM J Control Optim* 48:2901–2916
13. Eppler K, Harbrecht H (2010) Tracking the Dirichlet data in L^2 is an ill-posed problem. *J Optim Theory Appl* 145: 17–35
14. Eppler K, Harbrecht H (2012) On a Kohn-Vogelius like formulation of free boundary problems. *Comput Optim Appl* 52:69–85
15. Ben Abda A, Bouchon F, Peichl, GH, Sayeh M, Touzani R (2013) A Dirichlet-Neumann cost functional approach for the Bernoulli problem. *J Eng Math* 81:157–176
16. Bacani JB, Peichl GH (2013) On the first-order shape derivative of the Kohn-Vogelius cost functional of the Bernoulli problem. *Abstr Appl Anal* 2013 Article ID 384320 19 pp
17. Bacani JB, Peichl GH (2014) Solving the Exterior Bernoulli Problem using the Shape Derivative Approach. *Mathematics and Computing 2013*. International Conference in Haldia, India. In: Springer Proceedings in Mathematics and Statistics, vol 91. Springer XXII p 305
18. Rabago JFT, Bacani JB (2017) Shape optimization approach to the Bernoulli problem: a Lagrangian formulation. *IAENG Int J Appl Math* 47(4):417–424
19. Rabago JFT, Bacani JB (2018) Shape optimization approach for solving the Bernoulli problem by tracking the Neumann data: a Lagrangian formulation. *Commun Pur Appl Anal* 17(6):2683–2702
20. Ito K, Kunisch K, Peichl GH (2008) Variational approach to shape derivatives. *ESAIM Control Optim Calc Var* 14:517–539
21. Delfour MC, Zolésio J-P (2011) *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, 2nd ed. *Adv Des Control* 22 SIAM, Philadelphia
22. Delfour MC, Zolésio J-P (1988) Shape sensitivity analysis by min max differentiability. *SIAM J Control Optim* 26(4):834–862
23. Sokolowski J, Zolésio J-P (1992) *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer, Berlin

24. Ekeland I, Temam R (1976) Convex Analysis and Variational Problems. North-Holland Publishing Co Amsterdam. Translated from the French, Studies in Mathematics and its Applications vol. 1
25. Correa R, Seeger A (1985) Directional derivative of a minimax function. Nonlinear Anal Theory Meth Appl 9:13–22
26. Gilbarg D, Trudinger NS (1983) Elliptic Partial Differential Equations of Second Order. Springer-Verlag, Berlin
27. Kashiwabara T, Colciago CM, Dedè L, Quarteroni A (2015) Well-posedness, regularity, and convergence analysis of the finite element approximation of a generalized Robin boundary value problem. SIAM J Numer Anal 53(1):105–126
28. Bouchon F, Clain S, Touzani R (2005) Numerical solution of the free boundary Bernoulli problem using a level set formulation. Comput Method Appl Mech Eng 194:3934–3948
29. Flucher M, Rumpf M (1997) Bernoulli's free-boundary problem, qualitative theory and numerical approximation. J Reine Angew Math 486:165–204
30. Haddar H, Kress R (2016) A conformal mapping algorithm for the Bernoulli free boundary value problem. Math Meth Appl Sci 39:2477–2487
31. Kress R (2017) On Trefftz' integral equation for the Bernoulli free boundary value problem. Numer Math 136(2):503–522
32. Trefftz E (1916) Über die Kontraktion kreisförmiger Flüssigkeitsstrahlen. Z Math Phys 64:34–61
33. Harbrecht H, Mitrou G (2015) Stabilization of the trial method for the Bernoulli problem in case of prescribed Dirichlet data. Math Meth Appl Sci 38:2850–2863
34. Harbrecht H, Mitrou G (2014) Improved trial methods for a class of generalized Bernoulli problems. Math Anal Appl 420:177–194
35. Kuster CM, Gremaud PA, Touzani R (2007) Fast numerical methods for Bernoulli free boundary problems. SIAM J Sci Comput 29:622–634
36. Novruzi A, Roche J-R (2000) Newton's method in shape optimisation: a three-dimensional case. BIT Numer Math 40(1):102–120
37. Simon J (1989) Second variation for domain optimization problems. In: F. Kappel, K. Kunisch and W. Schappacher (eds) Control and estimation of distributed parameter systems, International Series of Numerical Mathematics, no 91. Birkhäuser pp 361–378.
38. Neuberger J (2010) In: Morel J-M, Teissier B (eds) Sobolev gradients and differential equations. Lecture Notes in Mathematics. Springer, Berlin
39. Azegami H, Kaizu S, Shimoda M, Katamine E (1997) Irregularity of shape optimization problems and an improvement technique. In: Hernandez S, Brebbia CA (eds) Computer aided optimization design of structures V. Computational Mechanics Publications. Southampton, pp 309–326
40. Mohammadi B, Pironneau O (2001) Applied Shape Optimization in Fluids. Oxford University Press Inc, New York
41. Azegami H (2016) Solution of shape optimization problem and its application to product design. Mathematical Analysis of Continuum Mechanics and Industrial Applications, In: Itou H, Kimura M, Chalupcký V, Ohtsuka K, Tagami D, Takada A (eds) Mathematics for Industry 26. Springer, Singapore, pp 83–98
42. Azegami H, Wu ZQ (1996) Domain optimization analysis in linear elastic problems: Approach using traction method. SME Int J Ser A 39(2):272–278
43. Azegami H, Shimoda M, Katamine E, Wu ZC (1995) A domain optimization technique for elliptic boundary value problems. In: Hernandez S, El-Sayed M, Brebbia CA (eds) Computer aided optimization design of structures IV, Structural Optimization. Computational Mechanics Publications, Southampton, pp 51–58.
44. Azegami H (1994) A solution to domain optimization problems. Trans of Japan Soc of Mech Engrs, Ser A 60:1479–1486 (in Japanese)
45. Henrot A, Pierre M (2018) Shape Variation and Optimization: A Geometrical Analysis. Tracts in Mathematics 28 European Mathematical Society, Zürich
46. Beurling A (1957) On free boundary problems for the Laplace equation, Seminars on analytic functions, vol 1. Institute of Advance Studies Seminars, Princeton, pp 248–263
47. Alt A, Caffarelli LA (1981) Existence and regularity for a minimum problem with free boundary. J Reine Angew Math 325:105–144
48. Henrot A, Shahgholian H (1997) Convexity of free boundaries with Bernoulli type boundary condition. Nonlinear Anal Theory Methods Appl 28(5):815–823
49. Henrot A, Shahgholian H (2000) Existence of classical solution to a free boundary problem for the p -Laplace operator: (II) the interior convex case. Indiana Univ Math J 49(1):311–323
50. Cardaliaguet P, Tahraoui R (2002) Some uniqueness results for the Bernoulli interior free-boundary problems in convex domains. Electron J Differ Equ 102:1–16
51. Hecht F (2012) New development in FreeFem++. J Numer Math 20(3-4):251–265