In *Applied and Numerical PDEs: Scientific Computing in Simulation, Optimization and Control and its Multiphysics Applications.*
W. Fitzgibbon, Y. Kuznetsov, P. Neittaanmäki, J. Periaux, and O. Pironneau, eds., Springer, 2010. Author's accepted manuscript.

A Unified Discrete–Continuous Sensitivity Analysis Method for Shape Optimization

Martin Berggren[1]

Department of Computing Science, Umeå University, Sweden
martin.berggren@cs.umu.se

Dedicated to Roland Glowinski

Summary. Boundary shape optimization problems for systems governed by partial differential equations involve a calculus of variation with respect to boundary modifications. As typically presented in the literature, the first-order necessary conditions of optimality are derived in a quite different manner for the problems before and after discretization, and the final directional-derivative expressions look very different. However, a systematic use of the material-derivative concept allows a unified treatment of the cases before and after discretization. The final expression when performing such a derivation includes the classical before-discretization ("continuous") expression, which contains objects solely restricted to the design boundary, plus a number of "correction" terms that involve field variables inside the domain. Some or all of the correction terms vanish when the associated state and adjoint variables are smooth enough.

Computer simulations of systems in science and engineering provide an efficient and cost effective tool to explore how performance depends on geometric features of the system components. An attractive alternative to trial-and-error testing is numerical design optimization, in which we introduce a parameterization of the geometry and let a numerical optimization algorithm interact with the simulation software in order to explore the parameter space. *Boundary shape optimization* is a strategy for design optimization that examines displacements of the boundary to a given domain. Such optimization is a powerful tool for final design, in order to put the final touch to a given configuration. Numerical boundary shape optimization typically uses body-fitted meshes, which makes the method suitable for problem exhibiting boundary layers or other phenomena with high sensitivity to boundary smoothness.

Besides boundary shape optimization, there are other, conceptually different techniques for design optimization that can handle much more general geometries than those generated by displacements of a given boundary; the term *topology optimization* is often used to highlight the generality. In the so-called *material distribution method* for topology optimization, it is coefficients of the governing partial differential equations discretized on a fixed mesh that are subject to optimization [2]. Such methods can generate arbitrarily complex geometries and are therefore suitable for preliminary design studies. The price for the generality is the limited resolution of the boundary geometry: typically, the boundary is represented using a staircase approximation, which is likely to cause problems in connection with boundary layers, for instance.

Conceptually, boundary shape optimization is a calculus of variation with respect to boundary modifications and traces its historical roots back to the works by, for instance, Newton, Lagrange, and Hadamard. The modern development was initiated in the early 1970s, mainly by the French school of numerical analysis, through researchers like Cea, Glowinski, and Pironneau. Although the field has developed and matured over the years, it is perhaps fair to say that the impact on science and engineering practice has been limited.

In contrast, the technique of optimal layout of a linearly elastic structures using the material distribution method for topology optimization has indeed had a noticeable impact on the design of mechanical components. There are commercial software packages available, for instance from Altair Engineering and FE-design, which are increasingly used for the design of mechanical components, particularly in the vehicle and aerospace industries. Boundary shape optimization is then used as a post processing tool for the layout obtained by topology optimization. However, boundary shape optimization is not much used for practical engineering design outside of such structural "sizing". One reason for the limited impact can be the complexity of managing a system for shape optimization: software for parameterization of shapes, mesh deformation, solvers, sensitivity analysis, and optimization needs to be developed and interfaced in an intricate way. Another reason is computational: solving a shape optimization problem takes often at least an order of magnitude longer time than a pure simulation. Because of the explosive development of hard- and software resources, these hurdles are likely to be overcome eventually. The recent appearance of several monograph dedicated to shape optimization [4, 6, 8, 10, 11, 12] is hopefully indicative of a revival.

The key to be able to treat shape optimization problems with a large number of design variables lies the use of gradient-based optimization methods and, in particular, in the use of adjoint equations to extract the directional derivatives. The experience collected through my own involvement in boundary shape optimization strongly indicates that the sensitivity information—directional derivatives of objective functions and constraints—needs to be very accurately computed in order for the optimization algorithms to fully converge. As was early on recognized, not the least by Roland Glowinski and his colleagues when developing shape optimization techniques, the processes of discretization and differentiation do not commute in general. That is, a discretization of the necessary conditions of optimality (differentiate-then-discretize, or the "continuous" approach) does not generally lead to the same expressions as when deriving the necessary conditions for the discretized optimization problem (discretize-then-differentiate, or the "discrete" approach). The latter strategy is more reliable in my experience, but may be difficult to effectuate in practice for complicated problems. Glowinski & He [7] and Gunzburger [8, §2.9], among many others, discuss and offer perspectives on this somewhat controversial issue.

A disturbing fact is that the two approaches often appear to be unrelated: the procedure for deriving the first-order necessary conditions in the undiscretized case is typically different from the one used in the discrete case, and the final expressions look very different. These problems may have contributed to the reason why there are very attempts to perform analysis of convergence and approximation errors for shape optimization problems. One of the few attempts reported in the literature are by Di Cesare, Pironneau, and Polak [5], [12, Ch. 6].

The present article shows that a systematic use of the material derivative allows a unified sensitivity analysis in the undiscretized and discretized cases. To minimize technical issues, the derivation will be made for a model elliptic problem and will be largely formal (without existence proofs for instance). However, the derivation will be made in a way that does not violate the regularity properties of the discrete problem. The final directional-derivative expression (45) (which appears to be new) contains the "continuous" expression plus a number of correction terms that are generally nonzero in the discrete case, but that vanish when the state and adjoint solutions are regular enough.

1 A potential flow model problem

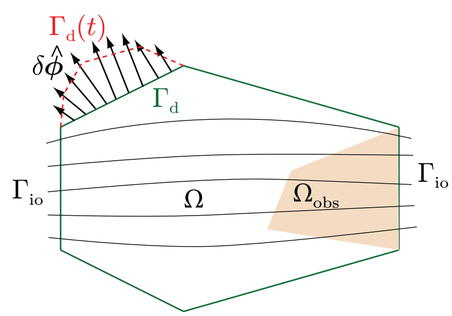

Fig. 1. An example domain for the model shape optimization problem

We consider the flow of an incompressible fluid in a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ with a Lipschitz boundary $\partial\Omega$ (figure 1). Fluid is flowing in and out through $\Gamma_{\mathrm{io}} \subset \partial\Omega$; otherwise there are impenetrable walls at the boundary. Let $\Gamma_{\mathrm{d}} \in \partial\Omega \setminus \Gamma_{\mathrm{io}}$ be a part of the boundary. We wish to manipulate the shape of the *design boundary* Γ_{d} in order to affect the velocity field in a desired way. Let $\mathscr{U}$ be the set of admissible design boundaries, whose definition may provide conditions such as bounds on curvature, bounds on displacements from a reference configuration, or requirements such as convexity of the domain. In order to perform a calculus of variation on Γ_{d}, we introduce a *design variation* $\delta\hat{\boldsymbol{\phi}} : \Gamma_{\mathrm{d}} \to \mathbb{R}^d$ that generates a family of deformed design boundaries $\Gamma_{\mathrm{d}}(t) \in \mathscr{U}$ in the following way: for each $\boldsymbol{x} \in \Gamma_{\mathrm{d}}$, there is an $\boldsymbol{x}(t) \in \Gamma_{\mathrm{d}}(t)$ such that

$$\boldsymbol{x}(t) = \boldsymbol{x} + t\,\delta\hat{\boldsymbol{\phi}}(\boldsymbol{x}) \quad t \in [0, \alpha]. \tag{1}$$

In order for formula (1) to generate Lipschitz design boundaries that are connected to the rest of the boundary, any feasible design variation needs to be Lipschitz continuous and vanishing on $\partial\Gamma_{\mathrm{d}}$. Any admissible $\delta\hat{\boldsymbol{\phi}}$ should also of course be compatible with the definition of $\mathscr{U}$. Further smoothness requirements on $\delta\hat{\boldsymbol{\phi}}$ will be introduced in §3 to allow differentiation. We assume that $\alpha > 0$ is small enough so that the mapping between Γ_{d} and $\Gamma_{\mathrm{d}}(t)$ is bijective for each $t \in [0, \alpha]$.

The displaced design boundaries $\Gamma_{\mathrm{d}}(t)$ generate a family of domains $\Omega(t)$ with Lipschitz boundaries. We consider the following potential-flow model defined on $\Omega(t)$:

$$\begin{aligned} -\Delta u + \epsilon u &= 0 \quad \text{in } \Omega(t), \\ \frac{\partial u}{\partial n} &= g \quad \text{on } \Gamma_{\mathrm{io}}, \\ \frac{\partial u}{\partial n} &= 0 \quad \text{on } \partial\Omega(t) \setminus \Gamma_{\mathrm{io}}, \end{aligned} \tag{2}$$

where $\epsilon > 0$ is a small "regularization" parameter introduced to avoid the singularity of the pure Neumann problem. The standard variational form of state equation (2) is:

$$\begin{aligned} &\text{Find } u(t) \in H^1\big(\Omega(t)\big) \text{ such that} \\ &\int_{\Omega(t)} \nabla v \cdot \nabla u(t) + \epsilon \int_{\Omega(t)} v u(t) = \int_{\Gamma_{\mathrm{io}}} v g \qquad \forall v \in H^1\big(\Omega(t)\big), \end{aligned} \tag{3}$$

where the notation $u(t)$ indicates the dependency on t.

Remark 1. Throughout this article, we will leave out symbols for volume and surface measure in the integrals, since the appropriate measures will be clear from the context.

Now introduce an *observation domain* Ω_{obs} that does not intersect with the design boundary; that is, $\Omega_{\mathrm{obs}} \subset \Omega$ such that $\overline{\Omega}_{\mathrm{obs}} \cap \overline{\Gamma}_{\mathrm{d}}(t) = \emptyset$. We wish to manipulate the shape of Γ_{d} such that the velocity field within the observation domain coincides as closely as possible with a given velocity field $\boldsymbol{u}_{\mathrm{obs}}$, a requirement that naturally leads to the objective function

$$J(\delta\hat{\boldsymbol{\phi}}; t) = \frac{1}{2} \int_{\Omega_{\mathrm{obs}}} |\nabla u(t) - \boldsymbol{u}_{\mathrm{obs}}|^2. \tag{4}$$

Some variation of the above problem is a common model problem for shape optimization in the context of fluid flow; Cesare et al. [5] consider essentially the same problem, for instance.

2 Sensitivity analysis

Here, we present the well-known formulas resulting from a sensitivity analysis of objective function (4), as described in the book by Pironneau [13], for instance. Section 2.1 gives the expressions before discretization, whereas corresponding expressions obtained after a finite-element discretization are reported in section 2.2.

2.1 Before discretization

A sensitivity analysis of objective function (4) and state equation (3) concerns the calculation of one-sided directional derivatives of the objective function with respect to design variation $\delta\hat{\boldsymbol{\phi}}$; we will use the notation

$$\delta J(\delta\hat{\boldsymbol{\phi}}) = \frac{\mathrm{d}^+}{\mathrm{d}t} J(\delta\hat{\boldsymbol{\phi}}; t)\big|_{t=0} = \lim_{t\to 0^+} \frac{J(\delta\hat{\boldsymbol{\phi}}; t) - J(\delta\hat{\boldsymbol{\phi}}; 0)}{t}. \tag{5}$$

The use of the *one-sided* derivative is essential when performing sensitivity analysis around admissible designs for which geometry constraints are active.

The classical expression for the directional derivative is

$$\delta J(\delta\hat{\boldsymbol{\phi}}) = -\int_{\Gamma_\mathrm{d}} \boldsymbol{n}\cdot\delta\hat{\boldsymbol{\phi}}\,\nabla u\cdot\nabla u^* - \epsilon\int_{\Gamma_\mathrm{d}} \boldsymbol{n}\cdot\delta\hat{\boldsymbol{\phi}}\,uu^*, \tag{6}$$

where $u^*\in H^1(\Omega)$ satisfies the adjoint equation

$$\int_\Omega \nabla w\cdot\nabla u^* + \epsilon\int_\Omega wu^* = \int_{\Omega_\mathrm{obs}} \nabla w\cdot(\nabla u - \boldsymbol{u}_\mathrm{obs}) \qquad \forall w\in H^1(\Omega). \tag{7}$$

Note the advantage of introducing the adjoint equation: the directional derivative for each feasible design variation $\delta\hat{\boldsymbol{\phi}}$ can be computed by repeated evaluation of integral (6) without solving any more equations.

Expression (6) is typically derived through a change of variables involving a smooth bijection between Ω and $\Omega(t)$. Such a mapping can be constructed by extending the boundary variation $\delta\hat{\boldsymbol{\phi}}$ to a domain variation $\delta\boldsymbol{\phi}:\overline{\Omega}\to\mathbb{R}^d$ such that for each point $\boldsymbol{x}\in\Omega$, there is a unique point $\boldsymbol{x}(t)\in\Omega(t)$ given by

$$\boldsymbol{x}(t) = \boldsymbol{x} + t\,\delta\boldsymbol{\phi}(\boldsymbol{x}), \tag{8}$$

and such that $\delta\boldsymbol{\phi}|_{\Gamma_\mathrm{d}} = \delta\hat{\boldsymbol{\phi}}$.

Although the extended mapping is used for the derivation, under certain smoothness assumptions of $\delta\boldsymbol{\phi}$ together with regularity properties that will be made explicit in section 5, it holds that the final expression (6) is independent of the particular choice of extension.

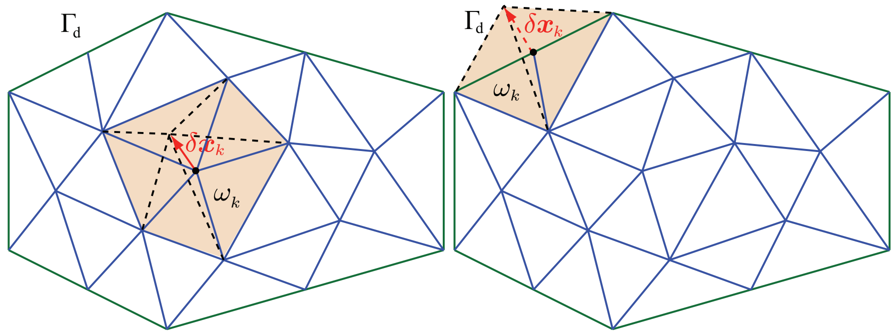

Fig. 2. Each mesh vertex displacement $t\,\delta\boldsymbol{x}_k$ is interpolated onto the support ω_k of the continuous piecewise-linear finite-element basis functions N_k^1.

2.2 After finite-element discretization

In the discrete case, it is natural to use the locations of the mesh vertices at the design boundary Γ_d as design variables. However, in order to retain mesh quality, it is in general necessary to modify the mesh inside the domain as well. Thus, for generality, associate with each mesh vertex a vector $\delta\boldsymbol{x}_k \in \mathbb{R}^d$ that indicates a feasible direction of movement for vertex k. Associated with *mesh vertex variation* $\delta\boldsymbol{x}_k$, it is convenient to define the domain variation $\delta\boldsymbol{\phi}_k = N_k^1\delta\boldsymbol{x}_k$, where N_k^1 is the continuous piecewise-linear finite-element basis function at vertex k. Subject to variation $\delta\boldsymbol{x}_k$, each $\boldsymbol{x}(t)$ in the deformed domain $\Omega(t)$ is then given by

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{x} + t\,\delta\boldsymbol{x}_k N_k^1(\boldsymbol{x}) \\ &= \boldsymbol{x} + t\,\delta\boldsymbol{\phi}_k(\boldsymbol{x}), \end{aligned} \tag{9}$$

and $\Omega(0) = \Omega$. Formula (9) interpolates deformation $t\,\delta\boldsymbol{x}_k$ at vertex k on the support of N_k^1. Note that the use of piecewise-linear basis functions implies that planar mesh surfaces and edges will remain planar under the deformation. Figure 2 illustrates deformation (9) in two cases. If the mesh on domain Ω is nondegenerate, then for each mesh vertex k, there is an $\alpha_k > 0$ such that the mesh associated with deformation (9) will also be nondegenerate for all $t \in [0, \alpha_k]$.

Now discretize equation (3) on the domain Ω using a conforming finite-element discretization in a subspace $V_h \subset H^1(\Omega)$. Given the deformation (9) associated with an arbitrary mesh vertex k, we may then define a family of discrete solutions

$$\begin{aligned} &u_h(t) \in V_h(t) \subset H^1(\Omega(t)) \text{ such that} \\ &\int_{\Omega(t)} \nabla v_h \cdot \nabla u_h(t) + \epsilon \int_{\Omega(t)} v_h u_h(t) = \int_{\Gamma_\mathrm{io}} v_h g \qquad \forall v_h \in V_h(t), \end{aligned} \tag{10}$$

and consider the discrete objective function

$$J_h(\delta\boldsymbol{\phi}_k;t) = \frac{1}{2}\int_{\Omega_{\mathrm{obs}}} |\nabla u_h(t) - \boldsymbol{u}_{\mathrm{obs}}|^2. \tag{11}$$

The following classical expression (e. g. [12, §6.5]) holds for the directional derivative of J_h:

$$\begin{aligned}\delta J_h = -\delta\boldsymbol{x}_k \cdot \int_\Omega \nabla N_k^1(\nabla u_h \cdot \nabla u_h^*) + \delta\boldsymbol{x}_k \cdot \int_\Omega \nabla u_h(\nabla u_h^* \cdot \nabla N_k^1) \\ +\delta\boldsymbol{x}_k \cdot \int_\Omega \nabla u_h^*(\nabla u_h \cdot \nabla N_k^1) - \epsilon\,\delta\boldsymbol{x}_k \cdot \int_\Omega u_h u_h^* \nabla N_k^1,\end{aligned} \tag{12}$$

where $u_h^* \in V_h$ such that

$$\int_\Omega \nabla w_h \cdot \nabla u_h^* + \epsilon \int_\Omega w_h u_h^* = \int_{\Omega_{\mathrm{obs}}} \nabla w_h \cdot (\nabla u_h - \boldsymbol{u}_{\mathrm{obs}}) \qquad \forall w_h \in V_h. \tag{13}$$

Expression (12) reveals expressions for the derivative of J_h with respect to variations of each mesh vertex in all coordinate directions (note that the integrals are vectors with d components). Once the state u_h and adjoint state u_h^* are known, all these derivatives can be computed by a single assembly loop over all elements. The derivatives can, for instance, be stored in a vector $\mathrm{D}J_h$ of dimension dn, where n is the total number of mesh vertices. Elements dk, $dk+1, \ldots, dk+d-1$ of $\mathrm{D}J_h$ then contains the d components of the integrals in expression (12).

However, in shape optimization, it does not make much sense to optimize the position of each mesh points independently. A good strategy is to modify the locations of the mesh vertices on Γ_{d} explicitly using updates from the optimization algorithm, and employ a mesh deformation strategy to move the rest of the mesh vertices indirectly in order to preserve mesh quality. In simple geometries, such a mesh deformation can be defined by an explicit formula based on the distance to Γ_{d}. A more general strategy, however, is to use a numerical deformation strategy, for instance based on elliptic smoothing [12, §5.3]. To describe the role of the mesh deformation in the derivative calculations, consider the spaces of discrete boundary and domain variations, $\widehat{\mathcal{U}}_h = \mathrm{span}(\delta\boldsymbol{\phi}_k)_{k\in\mathcal{V}(\Gamma_{\mathrm{d}})}$ and $\mathcal{U}_h = \mathrm{span}(\delta\boldsymbol{\phi}_k)_{k\in\mathcal{V}(\overline{\Omega})}$, where $\mathcal{V}(\gamma)$ denotes the set of mesh vertices located in the subdomain γ. A mesh deformation strategy defines a mapping $a : \widehat{\mathcal{U}} \to \mathcal{U}$, and the objective function that is in reality used for optimization when employing a mesh deformation is the composition $\widehat{J}_h = J_h \circ a$. By the chain rule, the derivative of mapping $\widehat{J}_h$ will be

$$\mathrm{D}\widehat{J}_h = \mathcal{A}^{\mathrm{T}}\,\mathrm{D}J_h, \tag{14}$$

where $\mathcal{A}$ is a matrix representation of the Jacobian of the mesh deformation mapping a.

Note that the discrete adjoint equation (13) constitutes a finite-element discretization of adjoint equation (7). However, the discrete directional derivative expressions (14), (12) carry no obvious resemblance to expression (6).

3 Shape and material derivatives of functions

In section 5, we will perform the sensitivity analysis in a way that simultaneously provides the seemingly quite different expressions (6) and (12). A main component of the derivation is the differentiation of state equations (3) and (10). There are two fundamentally different ways in which a function can be differentiated with respect to variations in the domain on which it is defined: as a *material* or as a *shape* derivative. These concepts, shortly reviewed below, are analogues to the material and spatial derivatives in continuous mechanics [9, §8]. For a thorough treatment of these concepts in the framework of shape optimization, see the monograph by Sokolowski and Zolesio [14]. This section aims to demonstrate a fact that seems curiously underappreciated in the shape-optimization literature: the material derivative is better suited, due to its favorable regularity properties, than the shape derivative for use in the sensitivity analysis.

We start by introducing the notation $\Omega(t) = \boldsymbol{\tau}_t(\Omega)$, where, for $\boldsymbol{x} \in \Omega$,

$$\boldsymbol{\tau}_t(\boldsymbol{x}) = \boldsymbol{x} + t\,\delta\boldsymbol{\phi}(\boldsymbol{x}), \qquad t \in [0, \alpha]. \tag{15}$$

For simplicity, we assume that the domain variation $\delta\boldsymbol{\phi}$ vanishes on Ω_{obs} and $\partial\Omega\setminus\Gamma_{\text{d}}$. For the problem before discretization, $\delta\boldsymbol{\phi}$ is an extension of $\delta\hat{\boldsymbol{\phi}}$ (which was defined solely on Γ_{d}) into a mapping from $\overline{\Omega}$ into $\mathbb{R}^d$. We require that the extended mapping is smooth enough so that the components of the second-order tensor $\nabla\delta\boldsymbol{\phi}$ are in $L^\infty(\Omega)$. In the discrete case, $\delta\boldsymbol{\phi}(\boldsymbol{x}) = \delta\boldsymbol{\phi}_k(\boldsymbol{x})$, where $\delta\boldsymbol{\phi}_k$ is given by expression (9) (here, $\delta\boldsymbol{\phi}$ can be made to vanish on Ω_{obs} and $\partial\Omega\setminus\Gamma_{\text{d}}$ by simply not considering any k for which corresponding mesh vertices are in Ω_{obs} or $\partial\Omega\setminus\Gamma_{\text{d}}$). By definition (9), it follows that the components of $\nabla\delta\boldsymbol{\phi}$ are in $L^\infty(\Omega)$ in the discrete case.

Now consider functions $p : \Omega(t)\times\mathbb{R} \to \mathbb{R}$. We will use $p(t)$ as a shorthand notation the for function $\boldsymbol{x} \mapsto p(\boldsymbol{x}, t)$.

Definition 1. *The* material derivative *of p with respect to domain variation $\delta\boldsymbol{\phi}$ at point $\boldsymbol{x} \in \Omega$ is*

$$\delta_m p(\boldsymbol{x}; \delta\boldsymbol{\phi}) = \lim_{t\to 0^+} \frac{p(\boldsymbol{\tau}_t(\boldsymbol{x}), t) - p(\boldsymbol{x}, 0)}{t} = \frac{\mathrm{d}^+}{\mathrm{d}t} p(\boldsymbol{\tau}_t(\boldsymbol{x}), t)\big|_{t=0},$$

provided that the pointwise limit exists.

(Whenever there is no risk for confusion, we will suppress the second argument and just use the notation $\delta_{\text{m}}p(\boldsymbol{x})$.) The material derivative is thus a (one-sided) derivative of the *compound* function $t \mapsto p(t)\circ\boldsymbol{\tau}_t$ (the "total" derivative). For $p(t)$ in a Banach space W, Definition 1 is easily extended to

$$\delta_{\text{m}} p = \frac{\mathrm{d}^+}{\mathrm{d}t} p(t)\circ\boldsymbol{\tau}_t\big|_{t=0} \tag{16}$$

with the limit in a Banach space $X \supset W$.

Definition 2. *The* shape derivative *of p with respect to design variation $\delta\boldsymbol{\phi}$ is the function*

$$\delta p = \delta_m p - \delta\boldsymbol{\phi} \cdot \nabla p(0) \tag{17}$$

Remark 2. Definition 2 imposes an a priori regularity difference between $\delta_{\mathrm{m}} p$ and δp due to the right-side gradient in expression (17). This difference is consistent with the typical behavior when differentiating the state variable in a shape optimization problem. As illustrated in examples (3) and (4) below, the material derivative of the state variable can typically be defined in the same space as the state variable itself, whereas the shape derivative typically cannot. An alternative definition of the shape derivative is as the partial derivative

$$\delta p(\boldsymbol{x}) = \lim_{t \to 0^+} \frac{p(\boldsymbol{x}, t) - p(\boldsymbol{x}, 0)}{t} \tag{18}$$

from which expression (17) follows by the chain rule applied on $\delta_{\mathrm{m}} p$. However, a complicating factor with definition (18) is that the two terms on the right side has different domains of definition, $\Omega(t)$ and Ω, respectively.

Following four examples highlight the different properties of the material and shape derivative.

Example 1. Let $g : \Omega \to \mathbb{R}$ be given. Define $p(t) = g \circ \boldsymbol{\tau}_t^{-1}$; that is, $p(\boldsymbol{x}, t)$ is defined by mapping back $\boldsymbol{x} \in \Omega(t)$ to corresponding point in Ω and evaluating g at the mapped-back point. Then

$$\begin{aligned} \delta_{\mathrm{m}} p &= \frac{\mathrm{d}^+}{\mathrm{d}t} \left(p(t) \circ \boldsymbol{\tau}_t\right)\big|_{t=0} = \frac{\mathrm{d}^+}{\mathrm{d}t} \left(g \circ \boldsymbol{\tau}_t^{-1} \circ \boldsymbol{\tau}_t\right)\big|_{t=0} = 0, \\ \delta p &= \delta_{\mathrm{m}} p - \delta\boldsymbol{\phi} \cdot \nabla p(0) = -\delta\boldsymbol{\phi} \cdot \nabla p(0). \end{aligned} \tag{19}$$

Thus, when $p(t)$ is "moving along" with the deformation, the material derivative vanishes. Next example illustrates the opposite situation.

Example 2. Let $f : \mathbb{R}^d \to \mathbb{R}$. Define $p(t) = f|_{\Omega(t)}$. Then

$$\begin{aligned} \delta_{\mathrm{m}} p &= \frac{\mathrm{d}^+}{\mathrm{d}t} \left(p(t) \circ \boldsymbol{\tau}_t\right)\big|_{t=0} = \frac{\mathrm{d}^+}{\mathrm{d}t} \left(f|_{\Omega(t)} \circ \boldsymbol{\tau}_t\right)\big|_{t=0} \\ &= \nabla f|_{\Omega(0)} \cdot \left(\frac{\mathrm{d}^+}{\mathrm{d}t} \boldsymbol{\tau}_t\right)\bigg|_{t=0} = \nabla f|_{\Omega(0)} \cdot \delta\boldsymbol{\phi} = \delta\boldsymbol{\phi} \cdot \nabla p(0), \\ \delta p &= \delta_{\mathrm{m}} p - \delta\boldsymbol{\phi} \cdot \nabla p(0) = 0. \end{aligned} \tag{20}$$

Thus, a function that is "fixed" with respect to the deformation yields a vanishing shape derivative, a property that is consistent with interpretation (18) of the shape derivative as a partial derivative.

Example 3. Let g belong to a finite element space $V_h \subset H^1(\Omega)$ such that $g(\boldsymbol{x}) = \sum_{k=1}^{N} g_k N_k^p(\boldsymbol{x})$, where N_k^p is a finite-element basis function that is globally continuous and whose restriction on each triangular or tetrahedral element is a polynomial of degree p. Define basis functions on the deformed domain $\Omega(t)$ by the expression $N_k^p(\boldsymbol{x}, t) = N_k^p(\boldsymbol{\tau}_t^{-1}(\boldsymbol{x}))$, as in example 1. The span of the functions $N_k^p(t)$ defines a family of finite-element spaces $V(t)$ on $\Omega(t)$. Each $p(t) \in V_h(t)$ may then be written

$$p(\boldsymbol{x}, t) = \sum_{k=1}^{N} p_k(t) N_k^p(\boldsymbol{x}, t). \tag{21}$$

As in example 1, we find that $\delta_{\mathrm{m}} N_k^p = 0$, $\delta N_k^p = -\delta\boldsymbol{\phi} \cdot \nabla N_k^p$ and thus

$$\begin{aligned} \delta_{\mathrm{m}} p &= \sum_{k=1}^{N} \delta p_k \, N_k^p, \\ \delta p = \delta_{\mathrm{m}} p - \delta\boldsymbol{\phi} \cdot \nabla p &= \sum_{k=1}^{N} \left(\delta p_k \, N_k^p - p_k \, \delta\boldsymbol{\phi} \cdot \nabla N_k^p \right), \end{aligned} \tag{22}$$

where

$$\delta p_k = \frac{\mathrm{d}^+}{\mathrm{d}t} p_k(t) \big|_{t=0}. \tag{23}$$

Note that $\delta_{\mathrm{m}} p \in V_h$ but $\delta p \notin V_h$! That is, the material derivative is *conforming* to the finite element space, whereas the shape derivative is not. Also note that the material derivative is obtained by differentiating only the coefficients of p (and not the basis functions) with respect to the deformation.

Example 4. Consider the solution $u(t) \in H^1(\Omega(t))$ to state equation (3). Sokolowski & Zolesio [14, §2.29] and Haslinger & Mäkinen [10, §2.5.2] discuss the existence of $\delta_{\mathrm{m}} p$ in similar situations, where they show that $\delta_{\mathrm{m}} u \in H^1(\Omega)$, provided that the domain deformations are sufficiently regular. As in example 3, the material derivative is defined in the same space as the state, but since $\delta u = \delta_{\mathrm{m}} u - \delta\boldsymbol{\phi} \cdot \nabla u$, the shape derivative typically has less regularity.

4 Rules for the material derivative

It is immediate from Definition 1 that the product rule holds for the material derivatives of functions f, g on $\Omega(t) \times \mathbb{R}$:

$$\delta_{\mathrm{m}}(fg) = \delta_{\mathrm{m}} f \, g + f \, \delta_{\mathrm{m}} g, \tag{24}$$

where, for simplicity of notation, we have suppressed the evaluations at zero: the right side should really be $\delta_{\mathrm{m}} f \, g(0) + f(0) \, \delta_{\mathrm{m}} g$. The rest of the article adheres to the same convention: for a function f on $\Omega(t) \times \mathbb{R}$, the symbol "$f$" outside a material derivative will denote its restriction to $t = 0$.

The shape derivative commutes with the spatial gradient, that is, $\delta\nabla = \nabla\delta$, but the material derivative does not: $\delta_{\mathrm{m}}\nabla \neq \nabla\delta_{\mathrm{m}}$. However, it holds that

$$\delta_{\mathrm{m}}(\nabla p) = \nabla(\delta_{\mathrm{m}} p) - (\nabla \delta \boldsymbol{\phi})^{\mathrm{T}} \nabla p, \tag{25}$$

or, in Cartesian components,

$$\delta_{\mathrm{m}}\left(\frac{\partial p}{\partial x_i}\right) = \frac{\partial}{\partial x_i}\delta_{\mathrm{m}} p - \sum_{j=1}^{d} \frac{\partial p}{\partial x_j} \frac{\partial}{\partial x_i} \delta \boldsymbol{\phi}_j, \quad i = 1, \ldots, d. \tag{26}$$

To prove expression (25), consider a finite-difference approximation of the material derivative:

$$\mathrm{D}_m^+ p(t) \stackrel{\mathrm{def}}{=} \frac{p(t) \circ \boldsymbol{\tau}_t - p(0)}{t}. \tag{27}$$

Differentiating both sides of expression (27) yields

$$\begin{aligned} \nabla \mathrm{D}_m^+ p(t) &= \frac{\nabla\big(p(t) \circ \boldsymbol{\tau}_t\big) - \nabla p(0)}{t} \\ &= \frac{1}{t}\left[\big(\boldsymbol{I} + t(\nabla \delta \boldsymbol{\phi})^{\mathrm{T}}\big) \nabla p(t) \circ \boldsymbol{\tau}_t - \nabla p(0)\right] \\ &= \frac{\nabla p(t) \circ \boldsymbol{\tau}_t - \nabla p(0)}{t} + (\nabla \delta \boldsymbol{\phi})^{\mathrm{T}} \nabla p(t), \end{aligned} \tag{28}$$

where the second equality follows from the chain rule applied on $\nabla\big(p(t) \circ \boldsymbol{\tau}_t\big)$ and from differentiation of $\boldsymbol{\tau}_t$ as defined in expression (15). Expression (28) implies that

$$\nabla(\delta_{\mathrm{m}} p) = \lim_{t \to 0} \nabla \mathrm{D}_m^+ p(t) = \delta_{\mathrm{m}} \nabla p + (\nabla \delta \boldsymbol{\phi})^{\mathrm{T}} \nabla p, \tag{29}$$

which is the expression we wanted to show.

The product rule (24) and expression (25) yields that

$$\delta_{\mathrm{m}}(\nabla q \cdot \nabla p) = \nabla \delta_{\mathrm{m}} q \cdot \nabla p + \nabla q \cdot \nabla \delta_{\mathrm{m}} p - \nabla q \cdot (\nabla_{\mathrm{S}}\, \delta \boldsymbol{\phi}) \nabla p, \tag{30}$$

where $\nabla_{\mathrm{S}}\, \delta \boldsymbol{\phi} = \nabla \delta \boldsymbol{\phi} + (\nabla \delta \boldsymbol{\phi})^{\mathrm{T}}$.

The rule for differentiating domain integrals that we will need in the following is [10, Lemma 3.3]

$$\delta\left(\int_{\Omega} f\right) = \frac{\mathrm{d}^+}{\mathrm{d}t}\left(\int_{\Omega(t)} f(t)\right)\Bigg|_{t=0} = \int_{\Omega} (\delta_{\mathrm{m}} f + f \nabla \cdot \delta \boldsymbol{\phi}). \tag{31}$$

Rules (25), (30), and (31) are the basic tools needed for a differentiation of the variational forms. Note that that there are no direct counterparts to expressions (25) and (30) for the shape derivative in the discrete case (when $p, q \in V_h$), and no shape-derivative counterpart to expression (31) with $f = \nabla q \cdot \nabla p$, since such expressions would involve second derivatives of the finite-element functions, which are not functions.

5 Sensitivity analysis using material derivatives

Equipped with the tools of sections 3 and 4, we now perform a derivation that simultaneously provides directional derivatives (6) and (12).

Let $V(t) \subset H^1\big(\Omega(t)\big)$ and define $V = V(0)$. In the case before discretization, $V(t) = H^1\big(\Omega(t)\big)$, whereas $V(t) = V_h(t)$, a finite-element space, in the discrete case. State equations (3) and (10) can then be written in the common form:

Let $u(t) \in V(t)$ such that

$$\int_{\Omega(t)} \nabla v(t) \cdot \nabla u(t) + \epsilon \int_{\Omega(t)} v(t)u(t) = \int_{\Gamma_{\mathrm{io}}} v(t)g \qquad \forall v(t) \in V(t), \tag{32}$$

and objective functions (4) and (11) in the form

$$j(\delta\boldsymbol{\phi}; t) = \frac{1}{2}\int_{\Omega_{\mathrm{obs}}} |\nabla u(t) - \boldsymbol{u}_{\mathrm{obs}}|^2. \tag{33}$$

Differentiating objective function (33) using differentiation rule (31) and observing that $\delta\boldsymbol{\phi}|_{\Omega_{\mathrm{obs}}} \equiv 0$ yields

$$\delta j(\delta\boldsymbol{\phi}) = \int_{\Omega_{\mathrm{obs}}} \nabla\delta_{\mathrm{m}} u \cdot (\nabla u - \boldsymbol{u}_{\mathrm{obs}}). \tag{34}$$

Differentiating state equation (32) at $t = 0$, using rules (24), (30), and (31) yields that

$$\begin{aligned} 0 = &\int_{\Omega} \left(\nabla\delta_{\mathrm{m}} v \cdot \nabla u + \epsilon\,(\delta_{\mathrm{m}} v)\, u\right) + \int_{\Omega} \left(\nabla v \cdot \nabla\delta_{\mathrm{m}} u + \epsilon\, v\, \delta_{\mathrm{m}} u\right) \\ &+ \int_{\Omega} \left(\nabla v \cdot \nabla u\, \nabla\cdot\delta\boldsymbol{\phi} + vu\, \nabla\cdot\delta\boldsymbol{\phi} - \nabla v \cdot (\nabla_{\mathrm{S}}\delta\boldsymbol{\phi})\, \nabla u\right) \end{aligned} \tag{35}$$

for each $v \in V$. Since $\delta_{\mathrm{m}} v \in V$ (cf. examples 3 and 4), the first integral in expression (35) vanishes due to state equation (32) evaluated at $t = 0$. Now let $u^* \in V$ satisfy the adjoint equation

$$\int_{\Omega} \nabla w \cdot \nabla u^* + \epsilon \int_{\Omega} wu^* = \int_{\Omega_{\mathrm{obs}}} \nabla w \cdot (\nabla u - \boldsymbol{u}_{\mathrm{obs}}) \qquad \forall w \in V. \tag{36}$$

By choosing $v = u^*$ in expression (35) and making use of equation (36) with $w = \delta_{\mathrm{m}} u$, expression (35) reduces to

$$\begin{aligned} 0 = &\int_{\Omega_{\mathrm{obs}}} \nabla\delta_{\mathrm{m}} u \cdot (\nabla u - \boldsymbol{u}_{\mathrm{obs}}) \\ &\quad + \int_{\Omega} \big(\nabla u^* \cdot \nabla u\, \nabla\cdot\delta\boldsymbol{\phi} + u^* u\, \nabla\cdot\delta\boldsymbol{\phi} - \nabla u^* \cdot (\nabla_{\mathrm{S}}\delta\boldsymbol{\phi})\, \nabla u\big), \end{aligned} \tag{37}$$

from which we conclude that expression (34) can be written

$$\delta j(\delta\boldsymbol{\phi}) = -\int_{\Omega} \big(\nabla u^* \cdot \nabla u\, \nabla\cdot\delta\boldsymbol{\phi} + \epsilon u^* u\, \nabla\cdot\delta\boldsymbol{\phi} - \nabla u^* \cdot (\nabla_{\mathrm{S}}\delta\boldsymbol{\phi})\, \nabla u\big). \quad (38)$$

Substituting $\delta\boldsymbol{\phi} = \delta\boldsymbol{x}_k\, N^1(\boldsymbol{x})$ into expression (38) yields the discrete expression (12).

In order to proceed further, we need to integrate expression (38) by parts in a way that respects the regularity properties of the involved functions. We will use a notation borrowed from the context of discontinuous Galerkin methods [1, §3]. Let $\mathcal{T}_h$ be the set of elements (triangles or tetrahedrons) in a triangulation of the domain Ω (that is, $\Omega(0)$). Note that the triangulation will be completely superficial, without any effect on the solution, in the case before discretization. Denote by $H^1(\mathcal{T}_h)$ the space of functions in $L^2(\Omega)$ whose restriction to each element $K \in \mathcal{T}_h$ is in $H^1(K)$ (functions in $H^1(\mathcal{T}_h)$ may however contain jump discontinuities between neighboring elements in the discrete case). Denote by Σ the union of the boundaries to all elements in the triangulation. Denote by $T(\Sigma)$ the space of traces of functions in $H^1(\mathcal{T}_h)$ on Σ; such traces are uniquely defined on the domain boundary $\partial\Omega$ but are in general double valued on the element boundaries $\Sigma_0 = \Sigma \setminus \partial\Omega$ interior to the domain. Consider two neighboring elements K_1 and K_2 that shares the surface (3D) or the edge (2D) $\sigma \in \Sigma_0$, and denote by $\boldsymbol{n}_1$ and $\boldsymbol{n}_2 = -\boldsymbol{n}_1$ the unit normals on σ that are outward directed with respect to K_1 and K_2, respectively. For $q \in H^1(\mathcal{T}_h)$, define jumps on σ by

$$[\![q]\!] = q\big|_{\partial K_1\cap\sigma}\boldsymbol{n}_1 + q\big|_{\partial K_2\cap\sigma}\boldsymbol{n}_2. \quad (39)$$

For $q \in H^1(\mathcal{T}_h)$ and $\boldsymbol{\psi} \in H^1(\Omega)^d$ hold the integration-by-parts formula

$$\int_{\Omega} \nabla\cdot\boldsymbol{\psi}\, q = -\sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{\psi}\cdot\nabla q + \int_{\partial\Omega} \boldsymbol{n}\cdot\boldsymbol{\psi}\, q + \int_{\Sigma_0} \boldsymbol{\psi}\cdot[\![q]\!]. \quad (40)$$

We will now apply formula (40) with $q = \nabla u^*\cdot\nabla u$ and $\boldsymbol{\psi} = \delta\boldsymbol{\phi}$. Note that $q \in H^1(\mathcal{T}_h)$ and $\boldsymbol{\psi} \in H^1(\Omega)^d$ hold for these choices: before discretization, $q|_K$ is smooth by internal regularity of equations (32) and (36), and $\boldsymbol{\psi} \in H^1(\Omega)^d$ by assumption; after discretization, $q|_K$ is polynomial, and $\boldsymbol{\psi}$ is in an $H(\Omega)^d$ conforming finite-element space. Using formula (40), the first term in the right side of expression (38) can be written

$$\begin{aligned} -\int_{\Omega} \nabla u^*\cdot\nabla u\, \nabla\cdot\delta\boldsymbol{\phi} = {} & \sum_{K\in\mathcal{T}_h} \int_K \delta\boldsymbol{\phi}\cdot\nabla(\nabla u^*\cdot\nabla u) \\ & - \int_{\Gamma_{\mathrm{d}}} \boldsymbol{n}\cdot\delta\boldsymbol{\phi}\, \nabla u^*\cdot\nabla u - \int_{\Sigma_0} \delta\boldsymbol{\phi}\cdot[\![\nabla u^*\cdot\nabla u]\!], \end{aligned} \quad (41)$$

where we have used that $\delta\boldsymbol{\phi}$ vanishes on $\partial\Omega\setminus\Gamma_{\mathrm{d}}$. Integration by parts on the second term in the right side of expression (38) yields

$$-\epsilon \int_{\Omega} u^* u \, \nabla \cdot \delta\boldsymbol{\phi} = \epsilon \int_{\Omega} \delta\boldsymbol{\phi} \cdot \nabla(u^* u) - \epsilon \int_{\Gamma_\mathrm{d}} \boldsymbol{n} \cdot \delta\boldsymbol{\phi} \, u^* u. \tag{42}$$

Substituting expressions (41) and (42) into expression (38) and collecting terms, we obtain

$$\begin{aligned} \delta j(\delta\boldsymbol{\phi}) = &- \int_{\Gamma_\mathrm{d}} \boldsymbol{n} \cdot \delta\boldsymbol{\phi} \, (\nabla u^* \cdot \nabla u + \epsilon \, u^* u) - \int_{\Sigma_0} \delta\boldsymbol{\phi} \cdot [\![\nabla u^* \cdot \nabla u]\!] \\ &+ \sum_{K \in \mathcal{T}_h} \int_K \big(\delta\boldsymbol{\phi} \cdot \nabla(\nabla u^* \cdot \nabla u) + \nabla u^* \cdot (\nabla_\mathrm{S} \delta\boldsymbol{\phi}) \, \nabla u + \epsilon \, \delta\boldsymbol{\phi} \cdot \nabla(u^* u)\big). \end{aligned} \tag{43}$$

The two first terms in the last integral in expression (43) can be written

$$\delta\boldsymbol{\phi} \cdot \nabla(\nabla u^* \cdot \nabla u) + \nabla u^* \cdot (\nabla_\mathrm{S} \delta\boldsymbol{\phi}) \, \nabla u = \nabla(\delta\boldsymbol{\phi} \cdot \nabla u^*) \cdot \nabla u + \nabla u^* \cdot \nabla(\delta\boldsymbol{\phi} \cdot \nabla u), \tag{44}$$

as shown by expanding in Cartesian components, for instance. Substituting expression (44) into expression (43) yields

$$\begin{aligned} \delta J = &- \int_{\Gamma_\mathrm{d}} \boldsymbol{n} \cdot \delta\boldsymbol{\phi} \, (\nabla u^* \cdot \nabla u + \epsilon \, u^* u) - \int_{\Sigma_0} \delta\boldsymbol{\phi} \cdot [\![\nabla u^* \cdot \nabla u]\!] \\ &+ \sum_{K \in \mathcal{T}_h} \int_K \big(\nabla(\delta\boldsymbol{\phi} \cdot \nabla u^*) \cdot \nabla u + \epsilon \, (\delta\boldsymbol{\phi} \cdot \nabla u^*) u\big) \\ &+ \sum_{K \in \mathcal{T}_h} \int_K \big(\nabla u^* \cdot \nabla(\delta\boldsymbol{\phi} \cdot \nabla u) + \epsilon \, u^* \delta\boldsymbol{\phi} \cdot \nabla u\big). \end{aligned} \tag{45}$$

Expression (45) contains, as its first term, the "continuous" directional derivative expression (6), but also three "correction" terms. The first correction term involves jumps of $\nabla u^* \cdot \nabla u$ at inter-element boundaries, whereas the second and third terms contain some particular weighted element-wise residuals of the state and adjoint equations, respectively, for which $\delta\boldsymbol{\phi} \cdot \nabla u^*$ and $\delta\boldsymbol{\phi} \cdot \nabla u$ replace the test functions. Some or all of these "correction terms" may vanish, depending on the situation:

Case 1 (before discretization). When $V = H^1(\Omega)$—the "continuous" case—the functions u and u^* are interior regular (and regular up to the boundary Γ_d when the boundary is smooth enough). In this case, the jump terms vanish due to the continuity of $\nabla u \cdot \nabla u^*$. Also, since $\delta\boldsymbol{\phi} \cdot \nabla u^* \in V$, $\delta\boldsymbol{\phi} \cdot \nabla u \in V$ in this case, the element residual terms will also vanish due to state and adjoint equations (32), (36). Hence, in this case, expression (45) reduces to the classic "continuous" expression (6).

Case 2 (lowest-order finite elements). If functions in V are linear on each element, the element residual terms vanish, since then $\nabla(\delta\boldsymbol{\phi} \cdot \nabla u^*)|_K = \nabla(\delta\boldsymbol{\phi} \cdot \nabla u)|_K \equiv 0$.

Case 3 (higher-order finite elements). Here, none of the terms vanishes in general, and expression (45) with $\delta\boldsymbol{\phi} = \delta\boldsymbol{x}_k\, N^1(\boldsymbol{x})$ just provides a different way of evaluating expression (12).

Case 4 (C^1 finite elements). When using the (rather unusual) class of C^1 finite elements (for instance the Argyris element [3, §3.2.10]), the inter-element jump terms vanish since then $[\![\nabla u^* \cdot \nabla u]\!] \equiv 0$.

Expression (45) links together the "discrete" expression (12) and the "continuous" expression (6) and constitutes therefore hopefully a first step in a rigorous numerical analysis of finite-element shape optimization. For instance, the convergence rate of the discrete Frechet derivative could be estimated by estimates of the jumps and residual terms that expression (45) exposes.

References

1. D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.
2. M. P. Bendsoe and O. Sigmund. *Topology Optimization. Theory, Methods, and Applications.* Springer, 2003.
3. S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer, New York, second edition, 2002.
4. D. Bucur and G. Buttazzo. *Variational Methods in Shape Optimization Problems.* Birkhäuser, 2005.
5. N. Di Cesare, O. Pironneau, and E. Polak. Consistent approximations for an optimal design problem. Technical Report R 98005, Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, 4, place Jussueu, 75252 Paris Cedex 05, France, 1998.
6. M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries. Analysis, Differential Calculus, and Optimization.* SIAM, Philadelphia, 2001.
7. R. Glowinski and J. He. On shape optimization and related issues. In J. Borggaard, J. Burns, E. Cliff, and S. Schreck, editors, *Computational Methods for Optimal Design and Control,* Proceedings of the AFOSR workshop on Optimal Design and Control, Arlington, Virginia, 1997, pages 151–179. Birkhäuser, 1998.
8. M. D. Gunzburger. *Perspectives in Flow Control and Optimization.* SIAM, Philadelphia, 2003.
9. M. E. Gurtin. *An introduction to continuum mechanics.* Academic Press, 2003.
10. J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization. Theory, Approximation, and Computation.* SIAM, Philadelphia, 2003.
11. E. Laporte and P. Le Tallec. *Numerical Methods in Sensitivity Analysis and Shape Optimization.* Birkhäuser, 2003.
12. B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids.* Oxford University Press, 2001.
13. O. Pironneau. *Optimal Shape Design for Elliptic Systems.* Springer, New York, 1984. In Springer Series in Computational Physics.
14. J. Sokolowski and J.-P. Zolesio. *Introduction to Shape Optimization. Shape Sensitivity Analysis.* Springer-Verlag, Berlin, 1992.