



HAL
open science

Shape and topology optimization

Grégoire Allaire, Charles Dapogny, François Jouve

► **To cite this version:**

Grégoire Allaire, Charles Dapogny, François Jouve. Shape and topology optimization. Andrea Bonito; Ricardo H. Nochetto. Geometric partial differential equations, part II, 22, 2021, Handbook of Numerical Analysis, 10.1016/bs.hna.2020.10.004 . hal-02496063

HAL Id: hal-02496063

<https://hal.science/hal-02496063>

Submitted on 2 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SHAPE AND TOPOLOGY OPTIMIZATION

G. ALLAIRE¹, C. DAPOGNY¹ AND F. JOUVE³

¹ *CMAP, Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France,*

² *Univ. Grenoble Alpes, CNRS, Grenoble INP¹, LJK, 38000 Grenoble, France,*

³ *Université de Paris, Laboratoire Jacques-Louis Lions (LJLL), F-75013 Paris, France.*

ABSTRACT. This chapter is an introduction to shape and topology optimization, with a particular emphasis on the method of Hadamard for appraising the sensitivity of quantities of interest with respect to the domain, and on the level set method for the numerical representation of shapes and their evolutions. At the theoretical level, the method of Hadamard considers variations of a shape as “small” deformations of its boundary; this results in a mathematically convenient and versatile notion of differentiation with respect to the domain, which has historically often been associated with “body-fitted” geometric optimization methods. At the numerical level, the level set method features an implicit description of the shape, which arises as the negative subdomain of an auxiliary “level set function”. This type of representation is well-known to be very efficient when it comes to describing dramatic evolutions of domains (including topological changes). The combination of these two ingredients is an ideal approach for optimizing both the geometry and the topology of shapes, and two related implementation frameworks are presented. The first and oldest one is a Eulerian shape capturing method, using a fixed mesh of a working domain in which the optimal shape is sought. The second and newest one is a Lagrangian shape tracking method, where the shape is exactly meshed at each iteration of the optimization process. In both cases, the level set algorithm is instrumental in updating the shapes, allowing for dramatic deformations between the iterations of the process, and even for topological changes. Most of our applicative examples stem from structural mechanics although some other physical contexts are briefly exemplified. Other topology optimization methods, like density-based algorithms or phase-field methods are also presented, at a lesser level of details, for comparison purposes.

CONTENTS

1. Introduction	2
2. Structural optimization problems: models and examples	5
2.1. The mechanical setting	5
2.2. A model shape and topology optimization problem	7
2.3. The representation of shapes: “discretize-then-optimize” and “optimize-then-discretize” paradigms	8
2.4. A look forward	9
3. Existence and non existence of optimal shapes: the homogenization effect	9
3.1. A non existence result	9
3.2. Remedies to non existence	13
4. Differentiation with respect to the domain: the boundary variation method of Hadamard	13
4.1. The method of Hadamard	14
4.2. Generalities about shape derivatives	15
4.3. Derivative of the solution to a PDE posed on the domain: Eulerian and Lagrangian derivatives	19
4.4. Shape derivative of PDE constrained shape functionals: the rigorous way	20

¹Institute of Engineering Univ. Grenoble Alpes

4.5.	Differentiation of an observable of a domain-dependent function: introduction of the adjoint state	22
4.6.	Shape derivative of PDE constrained shape functionals: C�ea’s formal method	25
4.7.	Optimization of the shape of the interface between two different media	30
5.	Shape optimization using Hadamard’s method: theoretical framework	35
5.1.	A basic shape gradient algorithm and some expected difficulties	35
5.2.	The Hilbertian extension-regularization procedure	36
5.3.	Constrained shape optimization	39
6.	“Body-fitted” geometric optimization	42
6.1.	Computational mesh and mesh deformation	42
6.2.	Avoiding mesh degeneracy in geometric methods	43
6.3.	A numerical example: the benchmark cantilever test-case	45
6.4.	Limitations and variants	45
7.	Eulerian shape capturing topology optimization via the level set method	46
7.1.	The level set method for tracking the motion of a domain	47
7.2.	Numerical algorithms for the level set method	48
7.3.	Shape and topology optimization using the level set method	52
7.4.	Numerical examples	54
7.5.	Coupling the level set shape optimization method with the topological derivative	61
7.6.	Achievements of the level set method and a few numerical difficulties	64
8.	A Lagrangian, level set based mesh evolution method for shape and topology optimization	64
8.1.	Presentation of the method	64
8.2.	Switching between the level set and meshed descriptions of shapes	65
8.3.	Numerical examples	66
9.	Shape optimization in different physical contexts and a few recent challenges	67
9.1.	Beyond linearized elasticity: applications to different physical contexts	67
9.2.	Geometric and manufacturing constraints	72
10.	A glimpse to other popular shape and topology optimization frameworks	76
10.1.	The homogenization method	76
10.2.	Density-based formulations	81
10.3.	Phase field methods	83
10.4.	An optimality criteria algorithm based on topological derivatives	84
11.	Perspectives for shape and topology optimization	85
11.1.	Shape and topology optimization in connection with additive manufacturing	86
11.2.	Robust shape and topology optimization under uncertainties	87
	References	88

1. INTRODUCTION

Shape optimization is an old, classical and ubiquitous field of research, which has inspired deep mathematical theories as well as numerous crucial applications in industry and engineering. Its dynamism has never been so intense as in the recent decades, where the increase in the cost of raw materials and the necessity to decrease energy consumption have made it more decisive than ever to optimize the produced components from the early stages of design, so that they fulfil their intended purpose for a minimum amount of constituent material. These needs, together with the rise of computational power and the development of advanced mathematical programming methods, made shape and topology optimization a very popular discipline in industrial design, notably in structural engineering, where it has successfully predicted conceptual designs with unprecedented abilities; see for instance Figure 1 (left). More generally, shape and topology optimization concepts are very influential in science. Of course, they are used in structural mechanics, including civil engineering or architecture (see [40] and Figure 1), in fluid mechanics [61, 86, 188], especially in connection with aerospace

applications or with the design of cooling devices [96], but also in the modelling of quantum chemistry phenomena [57], in the description of the “optimal”, complex patterns observed in biology, such as the internal structure of bones [151], in electromagnetism or photonics [120, 128], or in multi-purpose data processing issues such as image segmentation, shape detection and reconstruction; see for instance [35, 186] and [85] for an overview.



FIGURE 1. (Left) *Shape and topology optimization of a landing gear by an industrial software (courtesy of *ansys*) (right) The Qatar National Convention Center (architect: Arata Isozaki) has the typical outline of a topology optimized structure (licence Creative Commons)*

From the mathematical and algorithmic viewpoints, the perhaps first systematic shape optimization framework builds on the classical method of *shape sensitivity* or *boundary variation*, which goes back to the work of Hadamard [117]; see e.g. [5, 82, 118, 138, 140, 152, 172]. In a nutshell, the method of Hadamard amounts to evaluate the sensitivity of the considered functionals of the domain (objective or constraints) with respect to “small” perturbations of its boundary, then to obtain locally optimal shapes via successive deformations of an initial guess. It is a very general method which can handle a wide variety of physical models and objective functions, but it suffers from one major drawback: it is unable to account for changes in the topology of the initial guess (its number of holes in 2d), and the optimization strategy often ends up falling into local minima with poor performance.

From the 80’s, so-called “topology optimization” paradigms were developed precisely as attempts to circumvent this difficulty. Historically, the first of these was the homogenization method, whereby “designs” are described by a variable material density (which may take intermediate, “grayscale” values between those 0 and 1 indicating void or bulk structure), and a microstructure tensor accounting for the material properties at the microscopic scale. It was soon followed by variants featuring only a variable material density (and no microstructure tensor for the microscopic material properties), like the popular SIMP method (Solid Isotropic Material with Penalization), see e.g. [4, 45, 43, 65, 176]. Although these density methods were very successful, they also encountered difficulties of their own. In particular, they do not always produce a crisp “black-and-white” design (with only 0-1 values of the density) even after a suitable penalization of intermediate densities, leaving wide grayscale regions within the final design, with awkward physical interpretation. There has been a great deal of effort to try and improve this situation, based on the introduction of several density fields and so-called projection schemes (see the review article [168]), but not all researchers or practitioners are using these advanced features of the SIMP framework.

There was clearly some room for alternative shape and topology optimization methods; in this context, the level set method was introduced by various pioneering works [21, 22, 148, 163, 183]. The main idea behind the level set method is to come back to the Hadamard shape differentiation framework, with a whole new

implementation setting, namely that of front propagation or free boundary tracking by level set algorithms, as was proposed in the seminal work of Osher and Sethian [146, 162] to deal with motion by mean curvature. The level set algorithm is a versatile and computationally efficient numerical tool, which is widely used in many fields such as fluid mechanics, combustion theory, image processing, etc. (see the chapter [159] in the same book). Relying on an implicit description of shapes, it allows to account for dramatic changes in the course of the evolution and in particular, it naturally leaves the room for topological changes. The combination of these two ingredients, the Hadamard method for shape differentiation and level set front propagation, yields a very powerful shape and topology framework with many impressive achievements. With an obvious bias from the knowledge of the authors, this chapter deals almost exclusively with this level set method for shape and topology optimization although, admittedly, multiple frameworks are available to address such problems (see Section 10).

The overview of shape and topology optimization by the level set method contained in this chapter relies primarily on applications to the optimal design of mechanical structures. However, the introduced mathematical ingredients are relevant in other optimal design frameworks; likewise, many other physical contexts outside from the realm of mechanical structures could be tackled with similar techniques (see Section 9 for some examples). The presented shape and topology optimization frameworks combine one theoretical ingredient, the boundary variation method of Hadamard and the resulting notion of “shape derivative”, with a numerical tool, the level set method for representing shapes and their (arbitrarily large) deformations. As we have mentioned, the boundary variation method of Hadamard has long been associated with *geometric optimization*, where only the geometry of the boundary of shapes is optimized, and their topology is fixed. Such is indeed the case when a “classical” numerical representation of shapes is assumed, for instance when they are described by means of a set of physical parameters (curvature radii, thickness, length of a cavity, etc.), of a set of control points associated to a CAD (Computer-Aided Design) representation, or by a computational mesh. We shall see however that the method of Hadamard proves equally efficient in the context of *topology optimization*, when an adapted numerical representations of shapes is chosen, such as the level set method, or when shape sensitivity is coupled with, for instance, topological derivatives. Hence, with perhaps a little abuse, we shall refer to the methods discussed in this chapter as shape and topology optimization methods.

Two numerical implementations of the level set method are presented in this context. The first and oldest one [21, 22, 148, 163, 183] is a Eulerian *shape capturing* method, using a fixed mesh of a working domain in which the optimal shape is sought. The second and more recent one [13, 14] is a Lagrangian *shape tracking* method, where the shape is exactly meshed at each iteration of the optimization process. In both cases, the level set algorithm is instrumental in updating the shapes and allowing for possible topology changes.

This chapter is mainly intended as a user’s guide for the practical implementation of the aforementioned shape and topology optimization methods. Throughout our presentation, we assume only basic knowledge from the reader about classical topics such as differential calculus, functional analysis and scientific computing (notably about the finite element method). Along this line of thinking, the proofs of the needed mathematical results are provided (or sketched) only when the underlying arguments are important to deal with applications.

This chapter is organized as follows. In Section 2, we present the model mechanical setting at stake, as well as the shape and topology optimization problems under scrutiny. In Section 3, we discuss the issue of existence and non existence of a globally optimal shape. Notably, the homogenization effect is explained, which is the cause of many numerical artefacts in shape optimization practice. The next Section 4 is devoted to the boundary variation method of Hadamard, which is our main mathematical tool for evaluating how quantities of interest depend on the domain. Derivatives with respect to the domain are defined; their rigorous calculation is thoroughly detailed, and is complemented with the formal method of C ea, which conveniently allows to retrieve correct formulas while saving much technical burden. Section 5 sets the scene for the practical use of shape optimization, still staying at the theoretical level. We outline a generic shape gradient algorithm, to be specialized according to the numerical environment of the user. The multi-purpose Hilbertian framework is presented, which notably enables a simple and elegant treatment of constrained optimization problems. The numerical implementation of shape and topology optimization algorithms is detailed from Section 6, properly speaking. In there, we present the historical, popular and

intuitive *geometric optimization framework* where shapes are equipped with a mesh which is deformed according to the prescriptions of the shape gradient supplied by the method of Hadamard. The limitations of this algorithm motivate the introduction of the “classical” level set method for shape and topology optimization (which is an Eulerian *shape capturing* method): the latter is extensively discussed in Section 7, after a general presentation of the level set method for interface tracking. In Section 8, we discuss a recent avatar of the level set method for shape and topology optimization: introducing remeshing techniques in the workflow, it is possible to guarantee that an exact mesh of the optimized design be available at each stage of the process, thereby adding a “Lagrangian” flavor to this a priori Eulerian paradigm, while retaining the versatility and the robustness of the original level set method (it is therefore a Lagrangian *shape tracking* method). We present in Section 9 two applications of the level set based framework of this chapter beyond the rather “simple” physical context of linearly elastic structures, notably in nanophotonics, and in the field of interacting fluid-structure interaction systems. We also show that this framework lends itself to a convenient formulation of the geometric constraints imposed by fabrication processes via the signed distance function. In Section 10, we sketch a few other popular paradigms for shape and topology optimization, and notably the homogenization method, and the derived, formal, density-based methods (including the very popular SIMP method in mechanical engineering). Eventually, a few burning challenges faced by optimal design techniques are outlined in Section 11, namely taking into account fabrication constraints imposed by additive manufacturing process, and the need to devise robust optimal designs with respect to environmental uncertainties.

Before closing this introduction, let us mention that multiple educational articles devoted to various shape and topology optimization frameworks are available in the literature, illustrated with companion demonstration codes. The interested reader may consult [25, 76] when it comes to geometric optimization, [33, 166] about the SIMP method, and [24, 60, 126] about the level set method. We also refer to [168] for a quite exhaustive review of the most popular shape and topology optimization frameworks, and to [177] for a more specific focus on level set based methods.

2. STRUCTURAL OPTIMIZATION PROBLEMS: MODELS AND EXAMPLES

This section introduces the physical setting of our applications: that of mechanical structures. We present at first in Section 2.1 the system of linearized elasticity and the simpler Laplace equation. We then describe in Section 2.2 the shape and topology optimization problems considered throughout this chapter, and we discuss several instances of concrete interest. We outline in Section 2.3 some of the dilemmas posed by the implementation of such problems, before setting more specifically the scope of this chapter and the main notation used throughout in Section 2.4.

2.1. The mechanical setting

2.1.1. The linearized elasticity system

A mechanical structure in its reference (i.e. undeformed) configuration is described by a *shape* $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3 in practice), that is, a bounded, Lipschitz domain, whose boundary is decomposed into three disjoint pieces: $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma$, where

- the structure is attached on the region $\Gamma_D \neq \emptyset$,
- traction loads $g : \Gamma_N \rightarrow \mathbb{R}^d$ are applied on Γ_N ,
- the remaining region Γ is traction-free.

In addition, Ω is subjected to body forces (accounting e.g. for gravity) $f : \Omega \rightarrow \mathbb{R}^d$; see Figure 2. In this situation, the displacement $u_\Omega : \Omega \rightarrow \mathbb{R}^d$ of Ω is the unique solution to the linearized elasticity system:

$$(2.1) \quad \begin{cases} -\operatorname{div}(\sigma(u_\Omega)) = f & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \Gamma_D, \\ \sigma(u_\Omega)n = g & \text{on } \Gamma_N, \\ \sigma(u_\Omega)n = 0 & \text{on } \Gamma. \end{cases}$$

Here, $\sigma(u) : \Omega \rightarrow \mathbb{R}^{d \times d}$ is the *stress tensor* induced by a displacement field $u : \Omega \rightarrow \mathbb{R}^d$: for any unit vector $\xi \in \mathbb{R}^d$, the vector $\sigma(u)(x)\xi$ is the force felt by the face orthogonal to ξ of a small cube of material around

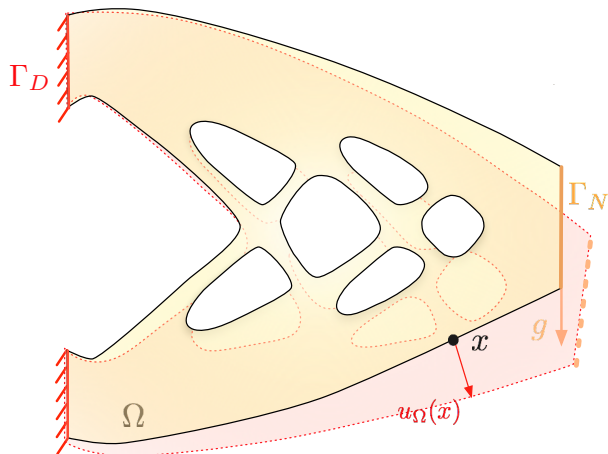


FIGURE 2. Example of a linearly elastic structure: a cantilever beam. The structure is represented in both reference and deformed configurations Ω (in yellow) and $(\text{Id} + u_\Omega)(\Omega)$ (in red), respectively.

x . The stress tensor $\sigma(u)$ is related to the strain tensor $e(u) := \frac{1}{2}(\nabla u + \nabla u^T)$ via the Hooke's law:

$$\sigma(u) = Ae(u),$$

where for any e in the set $\in \mathbb{R}_s^{d \times d}$ of symmetric $d \times d$ matrices,

$$Ae = 2\mu e + \lambda \text{tr}(e)\text{I},$$

I is the identity $d \times d$ matrix, and λ, μ are the Lamé parameters of the constituent material, satisfying $\mu > 0$ and $\lambda + 2\mu/d > 0$. The latter are often defined in terms of more physically relevant parameters:

$$\mu = \frac{E}{2(1 + \nu)}, \text{ and } \lambda = \frac{E\nu}{(1 + \nu)(1 + \nu(1 - d))},$$

where E is the Young's modulus, accounting for the resistance of the material to traction, and ν is the Poisson's ratio, measuring the transverse deflection of a piece of material when it is submitted to traction loads. We refer to [169] for a more complete introduction to the stakes of linearized elasticity.

As far as the mathematical setting is concerned, f belongs to the space $L^2(\Omega)^d$ of (vector-valued) square integrable functions on Ω , and $g \in L^2(\Gamma_N)^d$. The displacement u_Ω is an element of $H_{\Gamma_D}^1(\Omega)^d$, where

$$(2.2) \quad H_{\Gamma_D}^1(\Omega) := \{u \in H^1(\Omega), u = 0 \text{ on } \Gamma_D\}.$$

The system (2.1) has to be understood in the weak sense, where u_Ω is sought as the solution to the variational problem:

$$(2.3) \quad \text{Search for } u_\Omega \in H_{\Gamma_D}^1(\Omega)^d \text{ s.t. } \forall v \in H_{\Gamma_D}^1(\Omega)^d, \int_{\Omega} Ae(u_\Omega) : e(v) dx = \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g \cdot v ds,$$

whose well-posedness follows from the standard Lax-Milgram theory; see for instance [6].

2.1.2. The Laplace equation

On several occurrences in this chapter, our discussions will be simplified by replacing the previous linearized elasticity system with the following Laplace equation:

$$(2.4) \quad \begin{cases} -\text{div}(\gamma \nabla u_\Omega) = f & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \Gamma_D, \\ \gamma \frac{\partial u_\Omega}{\partial n} = g & \text{on } \Gamma_N, \\ \gamma \frac{\partial u_\Omega}{\partial n} = 0 & \text{on } \Gamma. \end{cases}$$

This model is involved in several different physical contexts; for instance:

- u is the temperature within Ω when the latter stands for a thermal chamber with conductivity γ , insulated from the outside on the region Γ of its boundary, and submitted to a heat source f and incoming heat flux g on Γ_N .
- u is the vertical displacement of an elastic body in a state of antiplane strain, i.e. only the component of the displacement which is orthogonal to a plane of interest does not vanish.

Mathematically, γ is a positive real number, $f \in L^2(\Omega)$ and $g \in L^2(\Gamma_N)$. The variational formulation of (2.4) is:

$$(2.5) \quad \text{Search for } u_\Omega \in H_{\Gamma_D}^1(\Omega) \text{ s.t. } \forall v \in H_{\Gamma_D}^1(\Omega), \int_{\Omega} \gamma \nabla u_\Omega \cdot \nabla v \, dx = \int_{\Omega} f v \, dx + \int_{\Gamma_N} g v \, ds.$$

Occasionally, we shall involve easy variants of this problem, where, for instance, one of the featured boundary conditions is omitted, a 0th order term is added, etc.

2.2. A model shape and topology optimization problem

The structural optimization problems considered in this chapter read:

$$(2.6) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } C(\Omega) \leq 0.$$

Here, \mathcal{U}_{ad} is the set of *admissible shapes* $\Omega \subset \mathbb{R}^d$. In our applications, an element $\Omega \in \mathcal{U}_{\text{ad}}$ is usually a “smooth” bounded domain (say, of class C^k , for some $k \geq 1$), whose boundary encloses a given region which is not subject to optimization. For instance, in the optimal design of elastic structures (see Section 2.1.1), the regions Γ_D, Γ_N where shapes are clamped, and where loads are applied, respectively, are often imposed by the context; they are fixed once and for all.

The minimized *objective function* $J : \mathcal{U}_{\text{ad}} \rightarrow \mathbb{R}$ is a measure of the performance of shapes. In structural optimization, $J(\Omega)$ typically depends on Ω via the elastic displacement u_Ω , solution to (2.1). Typical objective functions in this context are:

- The compliance

$$(2.7) \quad J(\Omega) = \int_{\Omega} Ae(u_\Omega) : e(u_\Omega) \, dx = \int_{\Omega} f \cdot u_\Omega \, dx + \int_{\Gamma_N} g \cdot u_\Omega \, ds,$$

which is equivalently the work of the external forces acting on Ω , or the energy dissipated inside the structure. Minimizing the compliance of Ω amounts to increasing its overall rigidity;

- a least-square difference

$$J(\Omega) = \int_{\Omega} k(x) |u_\Omega - u_T|^2 \, dx,$$

between the elastic displacement u_Ω and a target displacement $u_T : \Omega \rightarrow \mathbb{R}^d$, weighted by the localization factor $k(x)$;

- a p -norm of the stress tensor (as an approximation to the supremum norm)

$$J(\Omega) = \int_{\Omega} k(x) \|\sigma(u_\Omega)\|^p \, dx,$$

where $k(x)$ is again a localization factor, $p \geq 1$ is a real exponent, and $\|\cdot\|$ stands for the usual Frobenius norm over $d \times d$ matrices;

- (The negative of) the first resonance frequency of Ω :

$$J(\Omega) = -\lambda_\Omega,$$

where $\lambda_\Omega \geq 0$ is the first eigenvalue of (2.1), i.e. the smallest positive real number for which there exists $u \neq 0$ in $H_{\Gamma_D}^1(\Omega)^d$ satisfying:

$$\begin{cases} -\text{div}(\sigma(u)) = \lambda_\Omega u & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ \sigma(u)n = 0 & \text{on } \Gamma_N, \\ \sigma(u)n = 0 & \text{on } \Gamma. \end{cases}$$

In the formulation (2.6) of our shape and topology optimization problem, the function $C : \mathcal{U}_{\text{ad}} \rightarrow \mathbb{R}^p$ is a collection of p constraint functionals. Examples of such constraints are:

- The volume $\text{Vol}(\Omega)$ and the perimeter $\text{Per}(\Omega)$ of shapes:

$$(2.8) \quad \text{Vol}(\Omega) := \int_{\Omega} dx, \text{ and } \text{Per}(\Omega) = \int_{\partial\Omega} ds.$$

- The *minimum thickness* of Ω : thin features inside Ω should be controlled, since they are fragile and likely to break during fabrication.
- The *maximum thickness* of Ω : too thick regions should be avoided too, since they delay and hinder the cooling stage of the casting process.

These minimum and maximum thickness constraints are presented with a little more details in Section 9.2 below.

Remark 2.1.

- *One particular shape and topology optimization problem does not have a unique expression of the form (2.6): one may impose constraints on the optimized design by playing with the definition of the admissible set \mathcal{U}_{ad} , or alternatively as components of the functional $C(\Omega)$. Insofar as possible, in this chapter, we follow the convention that \mathcal{U}_{ad} encloses smooth enough shapes (which possibly contain a fixed subset), while volume, perimeter, and mechanical constraints are encoded in $C(\Omega)$.*
- *Similar examples of objective and constraint functionals can be thought off in the setting of the Laplace equation introduced in Section 2.1.2.*

2.3. The representation of shapes: “discretize-then-optimize” and “optimize-then-discretize” paradigms

One crucial issue in the treatment of (2.6) is the representation of the optimized design Ω , and the sense in which “derivatives” with respect to Ω are to be understood. Indeed, the *sensitivity* of the objective (or constraint) function with respect to the design is usually the cornerstone in the analysis of shape and topology optimization problems. From the theoretical viewpoint, it allows to write optimality conditions, while, from the practical point of view, it is the building block of most numerical algorithms, starting from the simplest gradient method.

There is unfortunately no “good” way to consider shapes and derivatives with respect to shapes in absolute terms: this essentially depends on the particular situation. One tempting approach, which is widely used in practice, starts with a discretization of the shape Ω into a finite set $\mathcal{P} = \{p_i\}_{i=1,\dots,N}$ of *design variables*. For instance, the p_i may account for physical properties of Ω (such as thicknesses, radii of curvature), for the control points of an associated CAD model, or for the vertices of a computational mesh; see Figure 3.

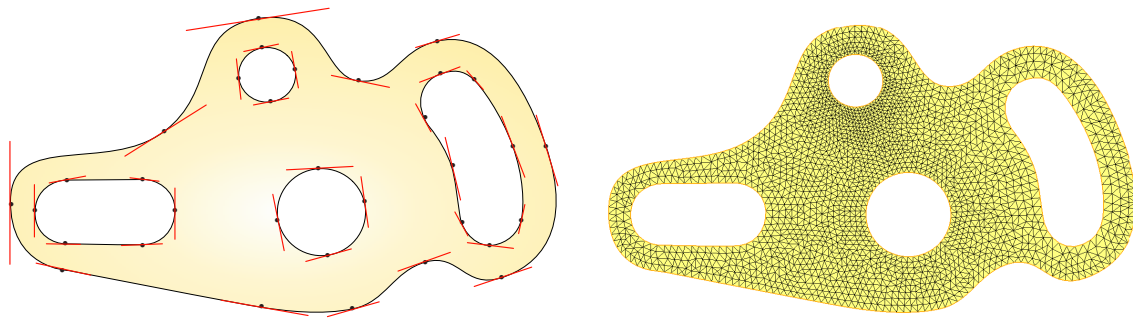


FIGURE 3. One domain may be supplied by a CAD representation (left), or by a computational mesh (right).

Doing so allows to reformulate (2.6) as:

$$\min_{\mathcal{P} \in \mathcal{P}_{\text{ad}}} J(\mathcal{P}) \text{ s.t. } C(\mathcal{P}) \leq 0,$$

where the objective and constraint functions $J(\mathcal{P})$ and $C(\mathcal{P})$ are now defined on a finite-dimensional space. Note that these functions may still depend on \mathcal{P} in a quite complicated way, e.g. via the solution to a partial differential equation whose parameters depend on \mathcal{P} . The sensitivity of J with respect to the parameters \mathcal{P} can however be evaluated by means of an adjoint technique similar to that presented in Section 4.5, in the continuous setting. This sensitivity can also be calculated thanks to automatic differentiation softwares; see Chap. 5 in [138] or [109]. The main asset of these “discretize-then-optimize” approaches (or direct methods in the language of optimal control) lies in their *consistency*: the derivatives used to update the design between optimization iterations are exactly those of the evaluated (discrete) functionals $J(\mathcal{P})$ and $C(\mathcal{P})$. Hence, this paradigm sticks to the discretization errors caused by the finite element model, such as those entailed by remeshing. As a result, the convergence of the process is in general smoother.

The “optimize-then-discretize” approach that we follow in this chapter goes the other way around. More precisely, we first introduce in Section 4 a continuous notion of derivative with respect to the domain; the values of $J(\Omega)$, $C(\Omega)$ and their shape derivatives are then calculated by using a suitable discretization of the resulting formulas (involving notably the finite element method for solving the partial differential equation for u_Ω , quadrature rules for evaluating integrals, etc.). This approach sticks to the theoretical, mathematical formulation of the problem; it tends to be more *accurate* with respect to the latter, insofar as the effective optimization trajectory is closer to the ideal, continuous one. We refer to [178] for a more exhaustive comparison of the “discretize-then-optimize” and “optimize-then-discretize” approaches.

2.4. A look forward

In this chapter, we first discuss some theoretical aspects of the shape and topology optimization problem (2.6). The question of existence of an optimal domain is broached in Section 3. The boundary variation method of Hadamard and the induced notion of differentiation with respect to the domain at the continuous level are presented in Section 4. From the next Section 5, we turn to the numerical implementation of these concepts.

Throughout this text, whenever it is possible, we rely on simplified versions of the optimization problem (2.6). In this spirit, we often study the constraint-free version

$$(2.9) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega).$$

Despite appearances, it is not a great loss of generality for numerical applications, since many constrained optimization algorithms somehow reduce the resolution of constrained problems such as (2.6) to that of a sequence of unconstrained problems of the form (2.9), as we explain in Section 5.3.

Besides, in most of the situations, the considered objective and constraint functionals arise under the form:

$$(2.10) \quad J(\Omega) = \int_{\Omega} j(u_\Omega) \, dx,$$

where $j : \mathbb{R}^d \rightarrow \mathbb{R}$ is a suitable function, and u_Ω , often referred to as the *state* is the solution to a partial differential equation posed on Ω , such as (2.1) or (2.4).

Finally, we are often not too specific about the nature of the set \mathcal{U}_{ad} of admissible designs.

3. EXISTENCE AND NON EXISTENCE OF OPTIMAL SHAPES: THE HOMOGENIZATION EFFECT

The perhaps first theoretical question regarding the analysis of a problem such as (2.6) is that of the existence or the non existence of an optimal design. It turns out that, “most” shape and topology optimization problems do not have a solution, at least under the form of a classical, “black-and-white” design. Even though the present chapter is oriented towards applications, we briefly discuss in this section the main obstruction to the existence of an optimal shape, the so-called *homogenization phenomenon*, since it has striking consequences, even at the numerical level. We illustrate this effect by an illuminating example in Section 3.1, before outlining in Section 3.2 a set of possible remedies to non existence in shape optimization.

3.1. A non existence result

We discuss an example whose complete details are tackled in [5]; the latter takes place in a slightly modified context from that of the Laplace equation introduced in Section 2.1.2. Let D be the unit square in \mathbb{R}^2 , and

let $\Omega \subset D$ be an arbitrary shape. Let u_Ω be the unique solution in $H^1(D)/\mathbb{R}$ (that is, up to constants) to the following two-phase conductivity equation inside D :

$$(3.1) \quad \begin{cases} -\operatorname{div}(\gamma_\Omega \nabla u_\Omega) = 0 & \text{in } D, \\ \gamma_\Omega \frac{\partial u_\Omega}{\partial n} = n_1 & \text{on } \partial D, \end{cases} \quad \text{where } \gamma_\Omega(x) = \begin{cases} \beta & \text{in } \Omega, \\ \alpha & \text{in } D \setminus \Omega, \end{cases}$$

and $0 < \alpha \leq \beta$ are positive constants. In the above, letting e_i be the i^{th} coordinate vector of \mathbb{R}^2 , $n_1 = n \cdot e_1$ is the horizontal coordinate of the normal vector to ∂D ; see Figure 4 for an illustration of this setting.

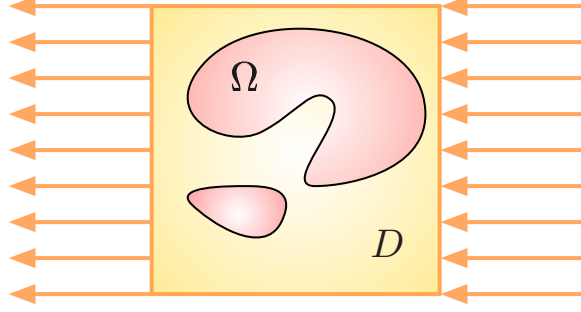


FIGURE 4. Setting of a shape optimization problem where optimal shapes fail to exist.

The following shape optimization problem is considered:

$$(3.2) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } \operatorname{Vol}(\Omega) = \eta|D|, \quad (|D| = 1),$$

where \mathcal{U}_{ad} is made of all measurable subsets $\Omega \subset D$, $\eta \in (0, 1)$ is an imposed volume fraction, and the objective function $J(\Omega)$ is the *compliance* of Ω :

$$J(\Omega) = \int_{\partial D} n_1 u_\Omega \, ds = \int_D \gamma_\Omega \nabla u_\Omega \cdot \nabla u_\Omega \, dx.$$

The main result of this section is the following.

Theorem 3.1. *The minimization problem (3.2) does not have a global solution.*

Sketch of proof. The proof proceeds in three steps: at first, we find a lower bound for the value of $J(\Omega)$, $\Omega \in \mathcal{U}_{\text{ad}}$. In a second step, we prove that this bound cannot be attained by a measurable subset $\Omega \in \mathcal{U}_{\text{ad}}$. Finally, we find a sequence Ω^n of admissible shapes such that $J(\Omega^n)$ converges to the lower bound. Before doing so, it will be convenient to rewrite $J(\Omega)$ as the minimum value of an Ω -dependent energy, defined for functions in $H^1(D)$. To this end, we rely on a convex duality argument, which is carefully detailed in [123].

Preliminary step: It follows from the Lax-Milgram theorem that the solution u_Ω to (3.1) is the unique solution (up to constants) to the following minimization problem:

$$(3.3) \quad J(\Omega) = -2 \min_{u \in H^1(D)} \left(\frac{1}{2} \int_D \gamma_\Omega \nabla u \cdot \nabla u \, dx - \int_{\partial D} n_1 u \, ds \right).$$

We now use the fact that, for any vector $\xi \in \mathbb{R}^2$, and any positive definite 2×2 matrix $A \in \mathbb{R}^{2 \times 2}$,

$$(3.4) \quad \frac{1}{2} A \xi \cdot \xi = \max_{\sigma \in \mathbb{R}^2} \left(\xi \cdot \sigma - \frac{1}{2} A^{-1} \sigma \cdot \sigma \right),$$

where the maximum is uniquely attained at $\sigma = A^{-1} \xi$. Then, (3.3) rewrites

$$(3.5) \quad J(\Omega) = \max_{u \in H^1(D)} \min_{\sigma \in L^2(D)^2} \mathcal{L}(u, \sigma),$$

where we have defined

$$\begin{aligned}\mathcal{L}(u, \sigma) &= -2 \int_D \sigma \cdot \nabla u \, dx + \int_D \gamma_\Omega^{-1} \sigma \cdot \sigma \, dx + 2 \int_{\partial D} n_1 u \, ds \\ &= 2 \int_D (\operatorname{div} \sigma) u \, dx - 2 \int_{\partial D} u \sigma \cdot n \, ds + \int_D \gamma_\Omega^{-1} \sigma \cdot \sigma \, dx + 2 \int_{\partial D} n_1 u \, ds.\end{aligned}$$

We know from what precedes that $(u, \sigma) = (u_\Omega, \gamma_\Omega \nabla u_\Omega)$ realizes the max-min in (3.5): it is therefore a saddle point for the functional $\mathcal{L}(u, \sigma)$, and so the min and max can be interchanged in (3.5):

$$J(\Omega) = \min_{\sigma \in L^2(D)^2} \max_{u \in H^1(D)} \mathcal{L}(u, \sigma).$$

Now, it is easily seen that for a given $\sigma \in L^2(D)^2$, one has:

$$(3.6) \quad \max_{u \in H^1(D)} \mathcal{L}(u, \sigma) = \begin{cases} \int_D \gamma_\Omega^{-1} \sigma \cdot \sigma \, dx & \text{if } \sigma \in \Sigma, \\ +\infty & \text{otherwise.} \end{cases}$$

where the set Σ is defined by:

$$\Sigma := \{ \sigma \in L^2(D)^2, \operatorname{div} \sigma = 0 \text{ in } D \text{ and } \sigma \cdot n = n_1 \text{ on } \partial D \}.$$

Indeed, if $\operatorname{div} \sigma$ does not vanish identically on D , there exists one function $\varphi \in C_c^\infty(D)$ such that $\int_D \operatorname{div} \sigma \varphi \, dx$ differs from 0 (say, is positive), and so

$$\max_{u \in H^1(D)} \mathcal{L}(u, \sigma) \geq \mathcal{L}(\lambda \varphi, \sigma) \xrightarrow{\lambda \rightarrow +\infty} +\infty.$$

A similar argument shows that $\sigma \cdot n$ must coincide with n_1 on ∂D for the maximum in (3.6) to retain a finite value. As a result of this discussion, $J(\Omega)$ has the expression:

$$J(\Omega) = \min_{\sigma \in \Sigma} \int_D \gamma_\Omega^{-1} \sigma \cdot \sigma \, dx.$$

First step: Lower bound for $J(\Omega)$. We remark that when $\Omega \subset D$ has measure $\operatorname{Vol}(\Omega) = \eta|D|$, the mean value of the conductivity γ_Ω in (3.1) over D is:

$$(3.7) \quad \gamma_0 := \frac{1}{|D|} \int_D \gamma_\Omega \, dx = (1 - \eta)\alpha + \eta\beta,$$

while the mean value of any $\sigma \in \Sigma$ is exactly e_1 , as a result of the following integration by parts, for $i = 1, 2$:

$$(3.8) \quad \left(\frac{1}{|D|} \int_D \sigma \, dx - e_1 \right) \cdot e_i = \frac{1}{|D|} \int_D (\sigma - e_1) \cdot \nabla x_i \, dx = \frac{1}{|D|} \int_{\partial D} (\sigma - e_1) \cdot n x_i \, ds - \frac{1}{|D|} \int_D (\operatorname{div} \sigma) x_i \, dx = 0.$$

We now rely on the fact that the function

$$\mathcal{I} : (0, \infty) \times \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \mathcal{I}(a, \sigma) = a^{-1} |\sigma|^2$$

has the following exact Taylor expansion about any point (a_0, σ_0) :

$$\mathcal{I}(a, \sigma) = \mathcal{I}(a_0, \sigma_0) + \mathcal{I}'(a_0, \sigma_0)(a - a_0, \sigma - \sigma_0) + \mathcal{I} \left(a, \sigma - \frac{a}{a_0} \sigma_0 \right),$$

where

$$\mathcal{I}'(a_0, \sigma_0)(a - a_0, \sigma - \sigma_0) = -\frac{|\sigma_0|^2}{a_0^2} (a - a_0) + \frac{2}{a_0} \sigma_0 \cdot (\sigma - \sigma_0),$$

as follows from a straightforward calculation. Hence, we obtain that for any measurable subset $\Omega \subset D$ with $\text{Vol}(\Omega) = \eta|D|$,

$$\begin{aligned}
 (3.9) \quad J(\Omega) &= \min_{\sigma \in \Sigma} \int_D \mathcal{I}(\gamma_\Omega, \sigma) \, dx \\
 &= \int_D \mathcal{I}(\gamma_0, e_1) \, dx + \min_{\sigma \in \Sigma} \int_D \mathcal{I} \left(\gamma_\Omega, \sigma - \frac{\gamma_\Omega}{\gamma_0} e_1 \right) \, dx \\
 &\geq \int_D \mathcal{I}(\gamma_0, e_1) \, dx \\
 &= |D| ((1 - \eta)\alpha + \eta\beta)^{-1}.
 \end{aligned}$$

where the second line follows from the mean values (3.7) and (3.8). This is the desired lower bound.

Second step: The lower bound is not attained. Note that in the process of deriving the above lower bound for $J(\Omega)$, the only place where equality is lost is the third line of (3.9). Equality holds in there if and only if

$$\mathcal{I} \left(\gamma_\Omega, \sigma - \frac{\gamma_\Omega}{\gamma_0} e_1 \right) = \gamma_\Omega^{-1} \left| \sigma(x) - \frac{\gamma_\Omega}{\gamma_0} e_1 \right|^2 = 0 \text{ for a.e. } x \in D.$$

In turn, this only happens when $\sigma = \frac{\gamma_\Omega}{\gamma_0} e_1$ a.e. in D , which is impossible since then σ would not satisfy $\sigma \cdot n = n_1$ on ∂D as is required from elements of Σ .

The inequality in (3.9) is therefore strict for any measurable domain $\Omega \subset D$ such that $\text{Vol}(\Omega) = \eta|D|$.

Third step: Construction of a minimizing sequence. We construct a sequence of measurable shapes $\Omega^n \subset D$ such that:

$$\text{Vol}(\Omega^n) = \eta|D|, \text{ and } J(\Omega^n) \rightarrow |D|((1 - \eta)\alpha + \eta\beta) \text{ as } n \rightarrow \infty.$$

Let indeed $\chi : (0, 1) \rightarrow \mathbb{R}$ be the function defined by:

$$\chi(t) = \begin{cases} 1 & \text{if } t \leq \eta, \\ 0 & \text{otherwise,} \end{cases}$$

and let Ω^n be the domain;

$$\Omega^n = \{(x_1, x_2) \in D, \chi(nx_2) = 1\};$$

roughly speaking, Ω^n is made of n regularly spaced horizontal strips with width $\frac{\eta}{n}$; see Figure 5.

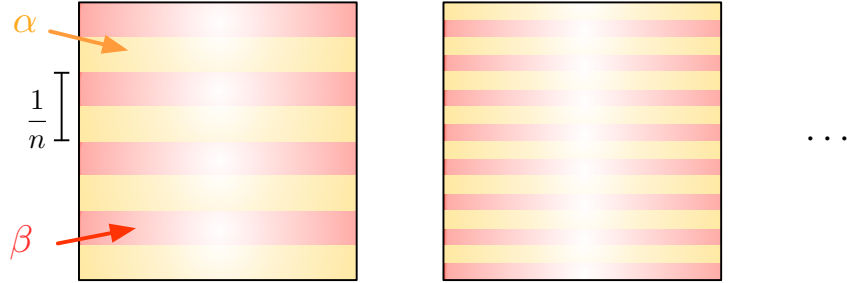


FIGURE 5. One minimizing sequence for $J(\Omega)$.

The mathematical theory of *homogenization* (see Section 10.1 for a glimpse and for instance [4] for a more exhaustive presentation) allows to prove that

$$(3.10) \quad u_{\Omega^n} \xrightarrow{n \rightarrow \infty} u^* \text{ weakly in } H^1(D),$$

where $u^* \in H^1(D)/\mathbb{R}$ is the solution to the *homogenized* problem:

$$\begin{cases} -\text{div}(A^* \nabla u^*) &= 0 & \text{in } D, \\ A^* \nabla u^* \cdot n &= n_1 & \text{on } \partial D, \end{cases}$$

and the homogenized matrix A^* reads:

$$A^* = \begin{pmatrix} (1-\eta)\alpha + \eta\beta & 0 \\ 0 & ((1-\eta)\alpha^{-1} + \eta\beta^{-1})^{-1} \end{pmatrix}.$$

The function u^* can be calculated in closed form; one verifies indeed that:

$$A^*\nabla u^* = e_1, \text{ and } u^*(x) = ((1-\eta)\alpha + \eta\beta)^{-1}x_1.$$

Finally, taking limits in the definition of $J(\Omega)$ with the help of the convergence (3.10) yields immediately:

$$\begin{aligned} \lim_{n \rightarrow \infty} J(\Omega^n) &= \int_{\partial D} n_1 u^* \, ds, \\ &= ((1-\eta)\alpha + \eta\beta)^{-1}, \end{aligned}$$

which is the desired value. □

3.2. Remedies to non existence

The above non existence result is quite instructive: roughly speaking, Theorem 3.1 reveals that it is beneficial for the considered objective to produce thinner and thinner patterns. An optimal design may not exist under the form of a classical “black-and-white” shape, but only as a *composite material*, described by a local density of the two materials α and β and a microstructure tensor, accounting for their local, microscopic arrangement. This *homogenization phenomenon* is the main cause of non existence of optimal shapes; see e.g. [51, 118].

There are classically two ways of modifying a shape and topology optimization such as (2.6) to enforce existence of optimal shapes:

- *Relaxation:* The set of admissible designs \mathcal{U}_{ad} is extended so that it include the microstructures produced by homogenization. This is the basic idea of the homogenization method, which is explained in [4, 65, 176]; see also Section 10.1 below for a brief presentation.
- *Restriction of the set of admissible designs:* The set of admissible designs \mathcal{U}_{ad} is restricted, so that the homogenization effect is prevented. For instance,
 - when a *uniform* regularity is imposed on admissible shapes $\Omega \in \mathcal{U}_{\text{ad}}$, such as a uniform cone condition or a uniform Lipschitz bound for their boundary, the set \mathcal{U}_{ad} enjoys a form of “compactness” which ensures the existence of optimal shapes therein [64].
 - In a similar spirit, adding a perimeter constraint to the considered optimization problem is another means to guarantee the compactness of minimizing sequences and so, existence of optimal shapes; see [26],
 - Limiting the number of holes of admissible shapes in 2d is yet another means to enforce existence of an optimal shape, as proved by Sverak [174] for a scalar problem, thanks to a capacity argument. This result was extended to the 2d linearized elasticity setting in [62].

Remark 3.1. *Even when the considered problem does not have a global solution, local solutions to (2.6) have great practical interest. Indeed, engineers often aim to find one design of their component close to, but better than the existing one, and not necessarily the optimal one, which could be too different and whose fabrication would require a complete change of the production line.*

4. DIFFERENTIATION WITH RESPECT TO THE DOMAIN: THE BOUNDARY VARIATION METHOD OF HADAMARD

In this section, we present the celebrated method of Hadamard for variations of shapes, and the resulting notion of differentiation with respect to the domain. This method was introduced by Hadamard in his seminal article [117] and it was extensively revisited in many further works. Here we follow the approach in [139, 140].

Following the reference books [5, 82, 118, 172], we start in Section 4.1 by defining properly the notion of shape derivative. We then provide in Section 4.2 a few “simple” examples, and we describe the general structure of shape derivatives of quite arbitrary functionals of the domain. We turn in Sections 4.3, 4.4 and 4.5 to the issue of differentiating the solution to a partial differentiation with respect to the domain, as well as an objective function in which it is involved. We next present in Section 4.6 the formal method of C ea, which allows for a simple and formal calculation of shape derivatives. The last Section 4.7 describes the

calculation of shape derivatives in the slightly different situation of two-phase problems, which conveniently allows to illustrate the previous techniques, and to highlight some dangers related to the practice of C ea’s method.

4.1. The method of Hadamard

Let $\Omega \subset \mathbb{R}^d$ be a bounded, Lipschitz open domain; we consider variations of Ω of the form:

$$(4.1) \quad \Omega_\theta := (\text{Id} + \theta)(\Omega), \text{ where } \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1,$$

where we recall that the Sobolev space $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, defined by

$$W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) := \{\theta \in L^\infty(\mathbb{R}^d)^d, \nabla\theta \in L^\infty(\mathbb{R}^d)^{d \times d}\}$$

is equivalently the Banach space of Lipschitz vector fields on \mathbb{R}^d with uniformly bounded derivatives; see [92]. Roughly speaking, Ω_θ in (4.1) is obtained from Ω by moving all its points according to the “small” vector field θ ; see Figure 6.

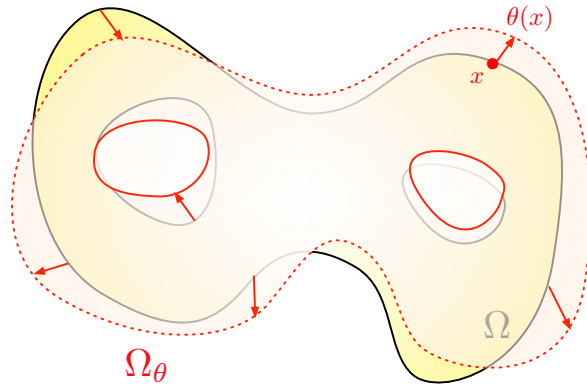


FIGURE 6. Variation Ω_θ of a shape $\Omega \subset \mathbb{R}^d$ according to a deformation field θ .

The following result shows in particular that when θ is “small enough”, Ω and its variation Ω_θ share the same topology; see Chap. 5 in [118].

Proposition 4.1. *For $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ with $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1$, the mapping $(\text{Id} + \theta)$ is a Lipschitz homeomorphism from \mathbb{R}^d into itself, with Lipschitz inverse.*

The previous definition of variations of a domain Ω naturally induces a notion of differentiation with respect to the domain.

Definition 4.1. *One function $J(\Omega)$ of the domain is said to be shape differentiable at Ω if the underlying mapping $\theta \mapsto J(\Omega_\theta)$, from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into \mathbb{R} is Fr chet differentiable at $\theta = 0$. The corresponding Fr chet derivative (or differential) is denoted by $J'(\Omega)(\theta)$ and the following expansion holds:*

$$J(\Omega_\theta) = J(\Omega) + J'(\Omega)(\theta) + o(\theta), \text{ where } \frac{o(\theta)}{\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)}} \xrightarrow{\theta \rightarrow 0} 0.$$

Remark 4.1. *A modern mathematical viewpoint is to see the set of shapes as a manifold, where $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ plays the role of tangent space; see [34, 160].*

As we have seen, shape optimization problems, of the form (2.6), usually bring into play a set \mathcal{U}_{ad} of admissible shapes. Accordingly, the deformations θ featured in the method of Hadamard are often restrained to a subspace Θ_{ad} of $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ (equipped with the relevant norm), so that variations Ω_θ of admissible shapes $\Omega \in \mathcal{U}_{\text{ad}}$, $\theta \in \Theta_{\text{ad}}$, remain admissible. Such a situations arises for instance when the boundary $\partial\Omega$ of shapes $\Omega \in \mathcal{U}_{\text{ad}}$ is required to contain a non optimizable region Σ : it suffices then to impose that

deformations $\theta \in \Theta_{\text{ad}}$ vanish on Σ . Likewise, when \mathcal{U}_{ad} contains shapes of class \mathcal{C}^k ($k \geq 1$), one usually considers as for the set Θ_{ad} :

$$\mathcal{C}^{k,\infty}(\mathbb{R}^d, \mathbb{R}^d) := \left\{ \theta \in \mathcal{C}^k(\mathbb{R}^d, \mathbb{R}^d), \sup_{|\alpha| \leq k} \|\partial^\alpha \theta\|_{L^\infty(\mathbb{R}^d)} < \infty \right\}.$$

To conclude this section, let us now place ourselves in the perspective of minimizing the considered functional of the domain $J(\Omega)$. As we shall discuss more extensively from Section 5, this task is usually conducted by using a *descent direction* for $J(\Omega)$, i.e. a deformation θ such that $J'(\Omega)(\theta) < 0$. This indeed implies from Definition 4.1 that for small $\tau > 0$:

$$J(\Omega_{\tau\theta}) = J(\Omega) - \tau J'(\Omega)(\theta) + o(\tau) < J(\Omega),$$

i.e. the value of $J(\Omega)$ is decreased when Ω is slightly deformed along θ .

4.2. Generalities about shape derivatives

In this section, we present general facts and methods about shape derivatives, which will be used repeatedly in this chapter. After recalling some classical material about differential and tangential calculus in Section 4.2.1, we provide in Section 4.2.2 the shape derivative of some “simple” shape functionals, before discussing in Section 4.2.3 the structure of the shape derivative of quite arbitrary functionals of the domain.

4.2.1. A primer in differential and tangential calculus

We first recall useful change of variables formulas in integrals on domains or their boundaries; see [92] §3.3.3 for the proof of the first point, and [118], Chap. 5, for that of the second point.

Proposition 4.2.

- Let $\Omega \subset \mathbb{R}^d$ be a bounded, Lipschitz domain, and let $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a Lipschitz invertible mapping, with Lipschitz inverse. Then, one measurable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ belongs to $L^1(T(\Omega))$ if and only if $f \circ T \in L^1(\Omega)$, and:

$$\int_{T(\Omega)} f \, dx = \int_{\Omega} |\det(\nabla T)| f \circ T \, dx.$$

- Let $\Omega \subset \mathbb{R}^d$ be a bounded domain of class \mathcal{C}^1 , and let T be a \mathcal{C}^1 diffeomorphism of \mathbb{R}^d . Then, one measurable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ belongs to $L^1(T(\partial\Omega))$ if and only if $f \circ T \in L^1(\partial\Omega)$, and:

$$\int_{T(\partial\Omega)} f \, dx = \int_{\partial\Omega} |\text{com}(\nabla T)n| f \circ T \, dx,$$

where $\text{com}(\nabla T)$ is the matrix of cofactors of ∇T .

As we shall see, the key ingredient in proving the differentiability of a function of the domain is the implicit function theorem, that we recall below for completeness; see [124], Chap. I, Th. 5.9.

Theorem 4.1. Let Θ, E, F be Banach spaces, $\mathcal{V} \subset \Theta$, $U \subset E$ be open sets. Let $p \geq 1$, and $\mathcal{F} : \mathcal{V} \times U \rightarrow G$ be a function of class \mathcal{C}^p . For a given point $(\theta_0, u_0) \in \mathcal{V} \times U$ where $\mathcal{F}(\theta_0, u_0) = 0$, assume that:

The (partial) differential $\frac{\partial \mathcal{F}}{\partial u}(\theta_0, u_0) : F \rightarrow G$ is a linear isomorphism.

Then there exists an open subset $\mathcal{V}_0 \subset \mathcal{V}$ of θ_0 in Θ , and a mapping $g : \mathcal{V}_0 \rightarrow U$ with the properties:

- (1) $g(\theta_0) = u_0$;
- (2) For $\theta \in \mathcal{V}_0$, the equation $\mathcal{F}(\theta, u) = 0$ has $u = g(\theta)$ for unique solution in E ;
- (3) The mapping g is of class \mathcal{C}^p .

We eventually collect a few useful notions of calculus on the boundary of domains, referring to Chap. 5 in [118] for further details. Till the rest of this section, Ω is a smooth bounded domain in \mathbb{R}^d , and n is the unit normal vector field to $\partial\Omega$, pointing outward Ω .

Definition 4.2.

- Let $V : \partial\Omega \rightarrow \mathbb{R}^d$ be a vector field. Its tangential part is the vector field $V_T : \partial\Omega \rightarrow \mathbb{R}^d$ given by:

$$V_T = V - (V \cdot n)n.$$

- Let $f : \partial\Omega \rightarrow \mathbb{R}$ be a function of class \mathcal{C}^1 . Its tangential gradient is the vector field $\nabla_T f : \partial\Omega \rightarrow \mathbb{R}^d$ defined by:

$$\nabla_T f = \nabla \tilde{f} - \frac{\partial \tilde{f}}{\partial n} n,$$

where \tilde{f} is any extension of f to an open neighborhood of $\partial\Omega$ ($\nabla_T f$ does not depend on the particular choice of such an extension).

- Let $V : \partial\Omega \rightarrow \mathbb{R}^d$ be a vector field of class \mathcal{C}^1 . Its tangential divergence is the function $\operatorname{div}_T V : \partial\Omega \rightarrow \mathbb{R}$ defined by:

$$\operatorname{div}_T V = \operatorname{div}(\tilde{V}) - \nabla \tilde{V} n \cdot n,$$

where \tilde{V} is any extension of V to an open neighborhood of $\partial\Omega$.

Remark 4.2. These definitions can be extended to larger classes of functions or vector fields, such as adequate Sobolev spaces on the boundary $\partial\Omega$. For instance, the tangential gradient $\nabla_T f$ is defined for functions $f \in H^1(\partial\Omega)$.

Introducing the mean curvature $\kappa = \operatorname{div} n$ of $\partial\Omega$ (note that with this definition, $\kappa(x)$ is positive at points $x \in \partial\Omega$ where Ω is locally convex), we now state a useful integration by parts formula on the boundary of Ω .

Proposition 4.3. Let $f : \partial\Omega \rightarrow \mathbb{R}$ and $V : \partial\Omega \rightarrow \mathbb{R}^d$ be respectively a function and a vector field of class \mathcal{C}^1 ; then:

$$\int_{\partial\Omega} \nabla_T f \cdot V \, ds = - \int_{\partial\Omega} f \operatorname{div}_T(V) \, ds + \int_{\partial\Omega} \kappa f V \cdot n \, ds.$$

4.2.2. First examples of shape derivatives

We first deal with the derivative of “simple” functionals of the domain Ω , insofar as they only depend on Ω via geometric entities and not via the solution u_Ω to a partial differential equation posed on Ω . These results are nevertheless key to all the forthcoming developments.

Theorem 4.2. Let $f \in W^{1,1}(\mathbb{R}^d)$, and consider the functional of the domain:

$$(4.2) \quad F(\Omega) = \int_{\Omega} f(x) \, dx.$$

Then $F(\Omega)$ is shape differentiable at any open set Ω , and its derivative reads:

$$(4.3) \quad \forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad F'(\Omega)(\theta) = \int_{\Omega} \operatorname{div}(f\theta) \, dx.$$

If in addition Ω is bounded and Lipschitz, the above rewrites:

$$(4.4) \quad F'(\Omega)(\theta) = \int_{\partial\Omega} f\theta \cdot n \, ds.$$

Hint of proof. The proof is representative of the usual techniques for proving the differentiability of a function with respect to the domain, and we outline the main ingredients; see [118], Chap. 5, for a complete presentation.

At first, a change of variables using Proposition 4.2 allows to transport the integral $F(\Omega_\theta)$ onto the fixed reference domain Ω :

$$F(\Omega_\theta) = \int_{\Omega} |\det(\mathbf{I} + \nabla\theta)| f \circ (\operatorname{Id} + \theta) \, dx.$$

We then prove the differentiability (and we calculate the derivatives) at $\theta = 0$ of the various terms in the above integrand:

- The mapping $\theta \mapsto |\det(\mathbf{I} + \nabla\theta)|$, from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into \mathbb{R} , is Fréchet differentiable at $\theta = 0$ and:

$$(4.5) \quad |\det(\mathbf{I} + \nabla\theta)| = 1 + \operatorname{div}(\theta) + o(\theta).$$

- The mapping $\theta \mapsto f \circ (\operatorname{Id} + \theta)$, from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $L^1(\mathbb{R}^d)$, is Fréchet differentiable at $\theta = 0$ and:

$$(4.6) \quad f \circ (\operatorname{Id} + \theta) = f + \nabla f \cdot \theta + o(\theta).$$

Now, using the product rule for derivatives, it follows that $\theta \mapsto F(\Omega_\theta)$ is differentiable at 0, and that its derivative is the volume form (4.3):

$$(4.7) \quad F'(\Omega)(\theta) = \int_{\Omega} (f \operatorname{div}(\theta) + \nabla f \cdot \theta) \, dx = \int_{\Omega} \operatorname{div}(f\theta) \, dx.$$

The surface expression (4.4) follows from an integration by parts, under the assumption that Ω is a bounded and Lipschitz domain. \square

It stems readily from this result that the volume $\operatorname{Vol}(\Omega)$ defined in (2.8) is shape differentiable at any bounded and Lipschitz domain Ω , and that:

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad \operatorname{Vol}'(\Omega)(\theta) = \int_{\partial\Omega} \theta \cdot n \, ds.$$

Theorem 4.2 is sometimes referred to as the *Transport Theorem*, or the *Reynold's Theorem*. It is the key ingredient in the derivation of the equations of motion in continuum mechanics from conservation principles; see [116] and Figure 7 (left) for an illustration.

The next result is devoted to the differentiation of surface integrals with respect to the domain. Its proof resembles much that of Theorem 4.2, and it is omitted; we refer again to [118], Chap. 5.

Theorem 4.3. *Let $g \in W^{2,1}(\mathbb{R}^d)$; then the functional*

$$G(\Omega) := \int_{\partial\Omega} g(x) \, ds(x).$$

is shape differentiable at any bounded domain Ω of class \mathcal{C}^2 when deformations θ are considered in $\mathcal{C}^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ and:

$$\forall \theta \in \mathcal{C}^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad G'(\Omega)(\theta) = \int_{\partial\Omega} \left(\frac{\partial g}{\partial n} + \kappa g \right) \theta \cdot n \, ds.$$

It follows that the perimeter $\operatorname{Per}(\Omega)$, defined in (2.8), is shape differentiable at any shape of class \mathcal{C}^2 , and:

$$\forall \theta \in \mathcal{C}^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad \operatorname{Per}'(\Omega)(\theta) = \int_{\partial\Omega} \kappa \theta \cdot n \, ds,$$

a result which in technical terms expresses that the mean curvature flow is the gradient flow associated to the minimization of the perimeter of shapes; see Figure 7 (right).

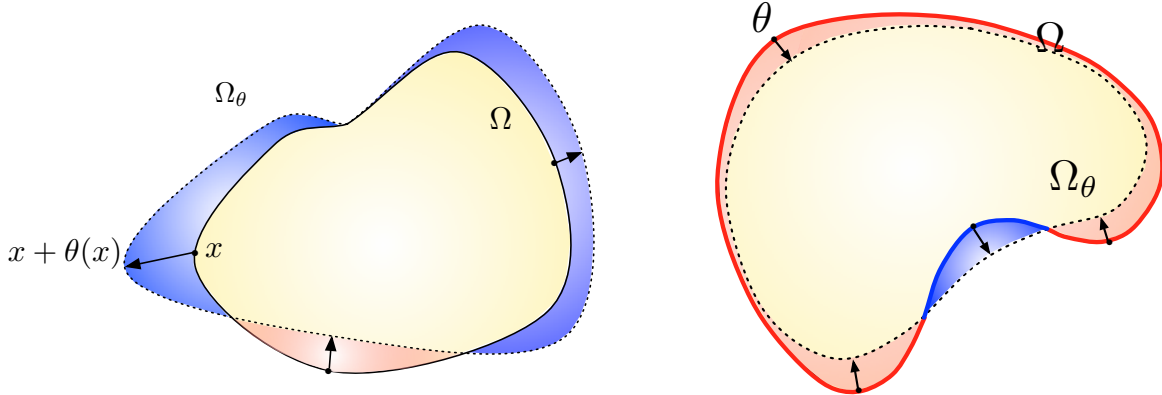


FIGURE 7. (Left) *Minimization of the integral (4.2): the deformation $\theta = -f(x)n$ satisfies $J'(\Omega)(\theta) < 0$; it makes Ω shrink in the (red) regions where $f(x) < 0$, and expand in the (blue) regions where $f(x) > 0$; (right) *minimization of the perimeter $\operatorname{Per}(\Omega)$: the deformation $\theta = -\kappa(x)n$ makes Ω shrink in the (red) positive curvature regions of $\partial\Omega$, and expand in the (blue) negative curvature regions.**

4.2.3. Structure of shape derivatives

The shape derivatives $J'(\Omega)(\theta)$ of the “simple” functionals $J(\Omega)$ considered in Section 4.2.2 depend only on the normal component $\theta \cdot n$ of the deformation θ on $\partial\Omega$. This fact is a quite general feature of shape derivatives, as stated in the next theorem (see Proposition 5.9.1 in [118]):

Theorem 4.4. *Let Ω be a domain of class C^1 and let $J : \mathcal{O}(\Omega) \rightarrow \mathbb{R}$ be a function of the domain defined over the set*

$$\mathcal{O}(\Omega) := \{(\text{Id} + \theta)(\Omega), \theta \in C^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \|\theta\|_{C^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1\}.$$

Assume that the mapping

$$C^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \ni \theta \mapsto J(\Omega_\theta) \in \mathbb{R}$$

is Fréchet differentiable at $\theta = 0$. Then the shape derivative $\theta \mapsto J'(\Omega)(\theta)$ is a distribution of order less than 1, which satisfies:

$$\forall \theta \in C^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \text{ s.t. } \theta \cdot n = 0 \text{ on } \partial\Omega, \quad J'(\Omega)(\theta) = 0.$$

This theorem is a rigorous statement of the fact that deforming Ω by a vector field θ taking tangential values on $\partial\Omega$ only results in a convection of Ω over itself. Hence, if $J(\Omega)$ has a “nice dependence” with respect to the domain, it holds $J(\Omega_\theta) \approx J(\Omega)$ (at first order), and so the shape derivative $J'(\Omega)(\theta)$ vanishes; see Figure 8.

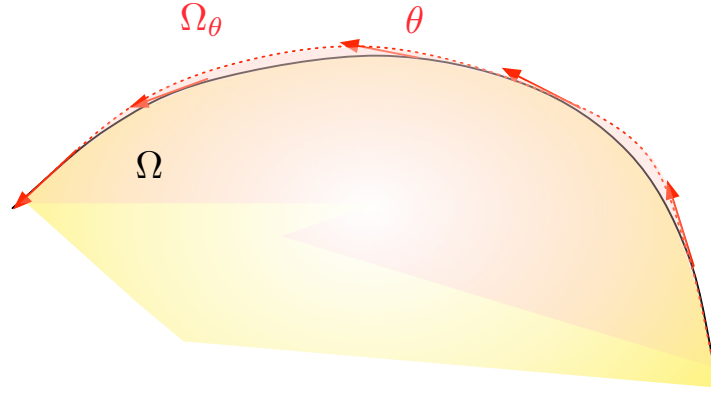


FIGURE 8. At first order, deforming Ω by a vector field θ which is tangential to $\partial\Omega$ does not alter the value of $J(\Omega)$.

In many applications, $J(\Omega)$ is an integral involving the solution u_Ω to a partial differential equation posed on Ω , as in (2.10), and the shape derivative $J'(\Omega)(\theta)$ turns out to have the more precise structure:

$$(4.8) \quad J'(\Omega)(\theta) = \int_{\partial\Omega} v_\Omega \theta \cdot n ds,$$

where $v_\Omega : \partial\Omega \rightarrow \mathbb{R}$ is a scalar field, depending on $J(\Omega)$; see for instance Theorem 4.6 below.

Finding a descent direction for $J(\Omega)$ is quite easy when its shape derivative $J'(\Omega)$ is of the form (4.8). Let us indeed define θ from its values on $\partial\Omega$:

$$(4.9) \quad \theta(x) = -v_\Omega(x)n(x) \text{ on } \partial\Omega.$$

It follows that

$$J'(\Omega)(\theta) = - \int_{\partial\Omega} v_\Omega^2 ds,$$

which is negative, unless the shape derivative $\theta \mapsto J'(\Omega)(\theta)$ vanishes identically, indicating that Ω is already a local minimum for J .

Note that it is also possible to find descent directions for $J(\Omega)$ even when its shape derivative is not readily available under the form (4.8). This operation is a bit more involved then, and it relies on the so-called Hilbertian extension-regularization procedure, that we describe in Section 5.2 below.

4.3. Derivative of the solution to a PDE posed on the domain: Eulerian and Lagrangian derivatives

We have hitherto discussed the differentiation with respect to the domain of “simple” functionals, depending rather explicitly on the domain Ω . In view of the targeted mechanical applications, we now present how to differentiate functionals depending on Ω via the solution u_Ω of a partial differential equation posed on Ω .

To set ideas, let us first focus on the case where the state $u_\Omega \in H_0^1(\Omega)$ is the unique solution to the Laplace equation with homogeneous Dirichlet boundary conditions on the whole $\partial\Omega$:

$$(4.10) \quad \begin{cases} -\Delta u_\Omega = f & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \partial\Omega. \end{cases}$$

We look for the shape derivative of a functional $J(\Omega)$ of the form

$$(4.11) \quad J(\Omega) = \int_\Omega j(u_\Omega) \, dx,$$

where $j : \mathbb{R} \rightarrow \mathbb{R}$ is of class C^2 and satisfies the growth conditions:

$$(4.12) \quad |j(x, u)| \leq C(1 + |u|^2), \quad |j'(x, u)| \leq C(1 + |u|), \quad \text{and } |j''(u)| \leq C.$$

The key point in the calculation of the shape derivative $J'(\Omega)$ is the definition of an appropriate notion of derivative for the mapping $\Omega \mapsto u_\Omega$. Formally speaking, there are actually two ways to achieve this, which are both illustrated in Figure 9.

The first, and perhaps most intuitive one is to proceed in a “Eulerian” fashion: for any fixed point $x \in \Omega$, we consider the derivative $u'_\Omega(\theta)(x)$ of the mapping:

$$(4.13) \quad \theta \mapsto u_{\Omega_\theta}(x).$$

The latter is well-defined from a neighborhood of 0 in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into \mathbb{R} since for small enough $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, x still belongs to the deformed shape Ω_θ , where u_{Ω_θ} takes its argument. On the contrary, for a boundary point $x \in \partial\Omega$, such a variation may be problematic if the vector field $\theta(x)$ is pointing inward Ω so that this point does not belong to Ω_θ or to its boundary.

The second approach is a “Lagrangian” one, inspired from mechanics, which relies on the *transported mapping*

$$\bar{u}_\Omega(\theta) := u_{\Omega_\theta} \circ (\text{Id} + \theta).$$

The latter not only makes sense in a point-to-point fashion (as (4.13) does), but also as a mapping from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $H_0^1(\Omega)$, and we may consider its Fréchet derivative $\theta \mapsto \mathring{u}_\Omega(\theta)$.

Let us finally note that, if both derivatives exist, the chain rule should entail, for any point $x \in \Omega$,

$$(4.14) \quad \mathring{u}_\Omega(\theta)(x) = \left. \frac{d}{d\theta} (u_{\Omega_\theta}(x + \theta(x))) \right|_{\theta=0} = u'_\Omega(\theta)(x) + \nabla u_\Omega(x) \cdot \theta(x).$$

As we are about to see, the second, “Lagrangian”, notion for the derivative of $\Omega \mapsto u_\Omega$ offers the most rigorous definition, and enjoys the best mathematical properties. We shall actually introduce the Eulerian derivative $u'_\Omega(\theta)$ only after the Lagrangian derivative, imposing the property (4.14) as a definition when it makes sense.

Definition 4.3. Let $\Omega \mapsto u_\Omega$ associate to any bounded, Lipschitz domain $\Omega \subset \mathbb{R}^d$ a function u_Ω in $H^1(\Omega)$.

- The mapping $\Omega \mapsto u_\Omega$ has a Lagrangian derivative at a particular shape Ω if the transported function

$$\theta \mapsto \bar{u}_\Omega(\theta) := u_{\Omega_\theta} \circ (\text{Id} + \theta),$$

defined from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $H^1(\Omega)$ is Fréchet differentiable at $\theta = 0$. Its Fréchet derivative $\theta \mapsto \mathring{u}_\Omega(\theta)$ is the Lagrangian derivative of u_Ω .

- The mapping $\Omega \mapsto u_\Omega$ has a Eulerian derivative $u'_\Omega(\theta)$ at Ω if it has a Lagrangian derivative, and if in addition $\nabla u_\Omega \in H^1(\Omega)$. One then defines the Eulerian derivative $u'_\Omega(\theta) \in H^1(\Omega)$ by:

$$u'_\Omega(\theta) = \mathring{u}_\Omega(\theta) - \nabla u_\Omega \cdot \theta.$$

Remark 4.3. These definitions can be extended to mappings $\Omega \mapsto u_\Omega$ where u_Ω belongs to another functional space than $H^1(\Omega)$, such as the Sobolev space $W^{s,p}(\Omega)$, $s \geq 0$, $p > 0$, or to a space of functions defined on the boundary $\partial\Omega$ such as $W^{s,p}(\partial\Omega)$.

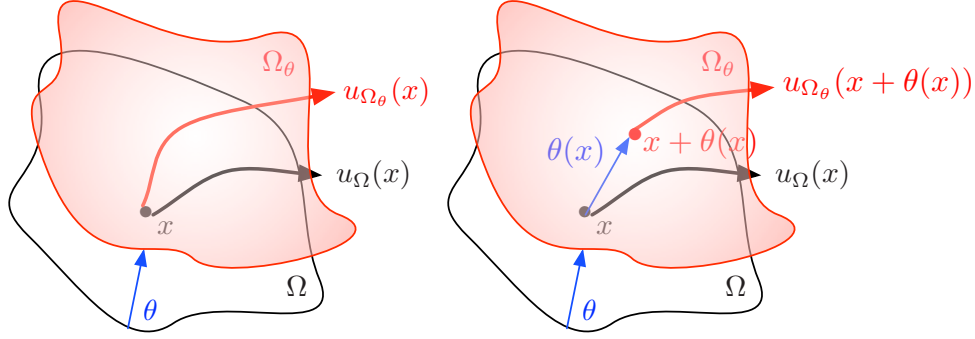


FIGURE 9. (Left) The Eulerian derivative $u'_\Omega(\theta)(x)$ is the derivative of the mapping $\theta \mapsto u_{\Omega_\theta}(x)$ for a fixed point $x \in \Omega$; (right) The Lagrangian derivative $u^\circ_\Omega(\theta)(x)$ is the derivative of the transported expression $\theta \mapsto u_{\Omega_\theta}(x + \theta(x))$.

4.4. Shape derivative of PDE constrained shape functionals: the rigorous way

In this section, we prove that the mapping $\Omega \mapsto u_\Omega$ which to a shape Ω associates the solution u_Ω to (4.10) has a Lagrangian derivative. The argument below is quite general and can be reused in multiple situations.

Theorem 4.5. *Assume that $f \in L^2(\mathbb{R}^d)$. Then, the solution $\Omega \mapsto u_\Omega$ of (4.10) has a Lagrangian derivative $u^\circ_\Omega(\theta)$ at any bounded Lipschitz shape Ω , which is the unique solution to the following variational problem:*

$$(4.15) \quad \forall v \in H_0^1(\Omega), \quad \int_\Omega \nabla u^\circ_\Omega(\theta) \cdot \nabla v \, dx = \int_\Omega \operatorname{div}(f\theta)v \, dx - \int_\Omega ((\operatorname{div}\theta)\mathbf{I} - (\nabla\theta + \nabla\theta^T)) \nabla u_\Omega \cdot \nabla v \, dx.$$

Sketch of the proof. The main ingredient is the implicit function Theorem 4.1.

1st step: We establish a variational formulation for the transported mapping $\overline{u}_\Omega(\theta) = u_{\Omega_\theta} \circ (\operatorname{Id} + \theta)$. To this end, we start from the variational formulation for u_{Ω_θ} :

$$\forall v \in H_0^1(\Omega_\theta), \quad \int_{\Omega_\theta} \nabla u_{\Omega_\theta} \cdot \nabla v \, dx = \int_{\Omega_\theta} f v \, dx.$$

We transport this formulation back to the fixed reference domain Ω by using the change of variables formula in Proposition 4.2:

$$\begin{aligned} \forall v \in H_0^1(\Omega_\theta), \quad \int_\Omega (\nabla u_{\Omega_\theta}) \circ (\operatorname{Id} + \theta) \cdot (\nabla v) \circ (\operatorname{Id} + \theta) |\det(\mathbf{I} + \nabla\theta)| \, dx = \\ \int_{\Omega_\theta} f \circ (\operatorname{Id} + \theta) v \circ (\operatorname{Id} + \theta) |\det(\mathbf{I} + \nabla\theta)| \, dx. \end{aligned}$$

Now, using the chain rule, we see that:

$$\nabla(u \circ (\operatorname{Id} + \theta)) = (\mathbf{I} + \nabla\theta^T)(\nabla u) \circ (\operatorname{Id} + \theta),$$

and so:

$$(4.16) \quad \forall v \in H_0^1(\Omega_\theta), \quad \int_\Omega A(\theta) \nabla(u_{\Omega_\theta} \circ (\operatorname{Id} + \theta)) \cdot \nabla(v \circ (\operatorname{Id} + \theta)) \, dx = \int_{\Omega_\theta} f \circ (\operatorname{Id} + \theta) v \circ (\operatorname{Id} + \theta) |\det(\mathbf{I} + \nabla\theta)| \, dx,$$

where the matrix $A(\theta)$ is defined by:

$$A(\theta) = |\det(\mathbf{I} + \nabla\theta)| (\mathbf{I} + \nabla\theta)^{-1} (\mathbf{I} + \nabla\theta)^{-T}.$$

Since (4.16) holds for arbitrary test functions $v \in H_0^1(\Omega_\theta)$, we may select v of the form $v = w \circ (\text{Id} + \theta)^{-1}$, where w varies over $H_0^1(\Omega)$. Doing so, it follows that the transported function $\overline{u_\Omega}(\theta) \in H_0^1(\Omega)$ is characterized by:

$$(4.17) \quad \forall w \in H_0^1(\Omega), \quad \int_{\Omega} A(\theta) \nabla(\overline{u_\Omega}(\theta)) \cdot \nabla w \, dx = \int_{\Omega_\theta} f \circ (\text{Id} + \theta) w |\det(\text{Id} + \nabla\theta)| \, dx.$$

2nd step: We prove the Fréchet differentiability of $\theta \mapsto \overline{u_\Omega}(\theta)$ by using the implicit function theorem. To this end, we rewrite the variational problem (4.17) as the search for a zero of a function $\mathcal{F} : W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \times H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$: for $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, $\overline{u_\Omega}(\theta)$ is the unique solution $u \in H_0^1(\Omega)$ to:

$$(4.18) \quad \mathcal{F}(\theta, u) = 0, \quad \text{where } \mathcal{F}(\theta, u) = \mathcal{A}(\theta, u) - b(\theta),$$

$\mathcal{A} : W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \times H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ is defined, for all $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ and $u \in H_0^1(\Omega)$, as the following element in $H^{-1}(\Omega)$:

$$\mathcal{A}(\theta, u) : H_0^1(\Omega) \ni v \mapsto \int_{\Omega} A(\theta) \nabla u \cdot \nabla v \, dx,$$

and $b : W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \rightarrow H^{-1}(\Omega)$ is defined by:

$$b(\theta) : H_0^1(\Omega) \ni v \mapsto \int_{\Omega_\theta} f \circ (\text{Id} + \theta) v |\det(\text{Id} + \nabla\theta)| \, dx.$$

It is now simple to verify that \mathcal{F} is of class \mathcal{C}^1 , and we wish to apply the implicit function Theorem 4.1 to the equation (4.18), at the point $(0, u_\Omega) \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \times H_0^1(\Omega)$. To this end, a simple calculation reveals that the partial derivative $\frac{\partial \mathcal{F}}{\partial u}(0, u_\Omega)$ is the mapping from $H_0^1(\Omega)$ into $H^{-1}(\Omega)$ given by:

$$\forall u \in H_0^1(\Omega), \quad \frac{\partial \mathcal{F}}{\partial u}(0, u_\Omega)(u) = \left[H_0^1(\Omega) \ni v \mapsto \int_{\Omega} \nabla u \cdot \nabla v \, dx \right] \in H^{-1}(\Omega),$$

and it follows immediately from the well-posedness of the variational formulation for (4.10) that it is an isomorphism. The implicit function Theorem thus ensures that there exists a function \mathcal{G} of class \mathcal{C}^1 , defined from a neighborhood U of 0 in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $H_0^1(\Omega)$, such that:

$$\forall \theta \in U, \quad \overline{u_\Omega}(\theta) = \mathcal{G}(\theta).$$

In particular, this proves that the mapping $\theta \mapsto \overline{u_\Omega}(\theta)$ is Fréchet differentiable in a neighborhood of $\theta = 0$.

3rd step: We calculate the derivative $\dot{u}_\Omega(\theta)$ by taking derivatives in the identity:

$$\mathcal{A}(\theta, \overline{u_\Omega}(\theta)) = b(\theta),$$

which is enabled by the result of the previous step. This yields:

$$\mathcal{A}(0, \dot{u}_\Omega(\theta)) = \frac{\partial b}{\partial \theta}(0)(\theta) - \frac{\partial \mathcal{A}}{\partial \theta}(0, u_\Omega)(\theta).$$

On the other hand, the above derivatives have the following explicit expressions, for arbitrary $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$,

$$\frac{\partial b}{\partial \theta}(0)(\theta) = \left[v \mapsto \int_{\Omega} (\nabla f \cdot \theta + f \operatorname{div}(\theta)) v \, dx \right],$$

and:

$$\frac{\partial \mathcal{A}}{\partial \theta}(0, u_\Omega)(\theta) = \left[v \mapsto \int_{\Omega} (\operatorname{div}(\theta) \nabla u_\Omega \cdot \nabla v - (\nabla\theta + \nabla\theta^T) \nabla u_\Omega \cdot \nabla v) \, dx \right].$$

As a result, the Lagrangian derivative $\dot{u}_\Omega(\theta)$ is characterized by the variational formulation (4.15). \square

We now turn to the existence of a Eulerian derivative for the mapping $\Omega \mapsto u_\Omega$, which holds true under slightly stronger assumptions. The following result follows from Theorem 4.5 by a straightforward (albeit a little tedious) calculation that we omit for brevity.

Corollary 4.1. *Under the assumptions of Theorem 4.5, and provided that in addition, the solution u_Ω to the state equation (4.10) belongs to $H^2(\Omega)$, the mapping $\Omega \mapsto u_\Omega$ in (4.10) has a Eulerian derivative $u'_\Omega(\theta)$ at Ω , which is the unique solution to the problem:*

$$(4.19) \quad \begin{cases} -\Delta u = 0 & \text{in } \Omega, \\ u = -\frac{\partial u_\Omega}{\partial n} \theta \cdot n & \text{on } \partial\Omega. \end{cases}$$

Remark 4.4. *The assumption that $u_\Omega \in H^2(\Omega)$ in Corollary 4.1 is pretty mild in practice. Indeed, it is a classical result from elliptic regularity theory that the solution u_Ω to (4.10) is smooth (say, in $H^{m+2}(\Omega)$ for $m \geq 0$) provided the domain Ω is smooth (say of class C^{m+2}) and f is smooth (say in $H^m(\Omega)$); see for instance [50], Chap. 9.*

Remark 4.5. *The formula (4.19) for the Eulerian derivative of u_Ω can be retrieved from heuristic considerations. Indeed, if $x \in \Omega$, then for θ sufficiently small, it holds $x \in \Omega_\theta$, and so $-\Delta u_{\Omega_\theta}(x) = 0$. Differentiating (formally) at $\theta = 0$ yields: $-\Delta u'_\Omega(\theta)(x) = 0$.*

Guessing the boundary condition associated to $u'_\Omega(\theta)$ in (4.19) is a little more involved. At first, using that, for all points $x \in \partial\Omega$, $u_{\Omega_\theta}(x + \theta(x)) = 0$, differentiation with respect to θ yields:

$$(4.20) \quad u'_\Omega(\theta)(x) + \nabla u_\Omega(x) \cdot \theta(x) = 0.$$

In addition, the boundary condition $u_\Omega = 0$ on $\partial\Omega$ implies that $\nabla u_\Omega(x) = \frac{\partial u_\Omega}{\partial n}(x)n(x)$; therefore, (4.20) simplifies into the desired formula:

$$u'_\Omega(\theta)(x) = -\frac{\partial u_\Omega}{\partial n}(x)(\theta \cdot n)(x).$$

4.5. Differentiation of an observable of a domain-dependent function: introduction of the adjoint state

Relying on the previous notion of derivative for the mapping $\Omega \mapsto u_\Omega$, we are in good position to calculate the derivative of the observable $J(\Omega)$ in (4.11).

4.5.1. A general derivative formula

Our first step is the following simple formula relating the shape derivative of $J(\Omega)$ to either the Lagrangian or the Eulerian derivative of the mapping $\Omega \mapsto u_\Omega$.

Proposition 4.4. *Let $\Omega \mapsto u_\Omega$ be a mapping associating a function $u_\Omega \in H_0^1(\Omega)$ to any bounded and Lipschitz domain Ω . If u_Ω has a Lagrangian derivative at Ω , the functional $J(\Omega)$ in (4.11) is shape differentiable at Ω , and:*

$$(4.21) \quad J'(\Omega)(\theta) = \int_\Omega (\operatorname{div} \theta j(u_\Omega) + j'(u_\Omega) u_\Omega^\circ(\theta)) \, dx.$$

If in addition, u_Ω has a Eulerian derivative at Ω , this rewrites:

$$(4.22) \quad J'(\Omega)(\theta) = \int_{\partial\Omega} j(u_\Omega) \theta \cdot n \, ds + \int_\Omega j'(u_\Omega) u'_\Omega(\theta) \, dx.$$

Proof. A change of variables based on Proposition 4.2 in the definition of $J(\Omega)$ yields, for sufficiently small $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$:

$$J(\Omega_\theta) = \int_\Omega |\det(\mathbf{I} + \nabla \theta)| j(\overline{u_\Omega}(\theta)) \, dx.$$

Taking derivatives in the above formula, using the first-order expansion (4.5) of the mapping $\theta \mapsto |\det(\mathbf{I} + \nabla \theta)|$ and the Definition 4.3 of the Lagrangian derivative, we readily obtain (4.21).

Rearranging this formula yields:

$$J'(\Omega)(\theta) = \int_\Omega \operatorname{div}(j(u_\Omega)\theta) \, dx + \int_\Omega j'(u_\Omega)(u_\Omega^\circ(\theta) - \nabla u_\Omega \cdot \theta) \, dx.$$

Formula (4.22) then follows from an integration by parts and the definition of the Eulerian derivative $u'_\Omega(\theta)$ (see again Definition 4.3). \square

4.5.2. Introduction of the adjoint state

The derivative formulas supplied by Proposition 4.4 are a little ill-suited from the practical point of view. Indeed, as we have anticipated in Section 4.1, our main interest is to find descent directions θ for $J(\Omega)$, that is, deformation fields θ such that $J'(\Omega)(\theta) < 0$. This task is not simple from the formulas supplied by Proposition 4.4: the calculation of $J'(\Omega)(\theta)$ for a particular deformation θ requires the resolution of a partial differential equation for the Lagrangian or the Eulerian derivative $u_\Omega^\circ(\theta)$ or $u'_\Omega(\theta)$ (see (4.15) and (4.19)), where θ plays the role of a parameter. It would prove considerably simpler if, on the contrary, $J'(\Omega)(\theta)$ were to enjoy a completely explicit dependence with respect to θ , as in the simple structure (4.8). Fortunately, such a reformulation can be achieved, up to the introduction of an auxiliary, *adjoint state* p_Ω . This general idea was originally developed by Pontryagin in the context of optimal control involving ordinary differential equations as state equations [154], and later by Lions for optimization problems under constraints taking the form of partial differential equations [130].

Proposition 4.5. *For any bounded Lipschitz domain Ω , the shape functional $J(\Omega)$ in (4.11) is shape differentiable, and the derivative reads:*

$$(4.23) \quad J'(\Omega)(\theta) = \int_{\Omega} \operatorname{div} \theta j(u_\Omega) \, dx + \int_{\Omega} ((\operatorname{div} \theta) \mathbf{I} - (\nabla \theta + \nabla \theta^T)) \nabla u_\Omega \cdot \nabla p_\Omega \, dx - \int_{\Omega} \operatorname{div}(f\theta) p_\Omega \, dx,$$

where the adjoint state p_Ω is the unique solution in $H_0^1(\Omega)$ to the equation:

$$(4.24) \quad \begin{cases} -\Delta p_\Omega = -j'(u_\Omega) & \text{in } \Omega, \\ p_\Omega = 0 & \text{on } \partial\Omega. \end{cases}$$

Proof. We know from Proposition 4.4 that $J(\Omega)$ is shape differentiable at Ω , with derivative:

$$(4.25) \quad \forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad J'(\Omega)(\theta) = \int_{\Omega} (\operatorname{div} \theta j(u_\Omega) + j'(u_\Omega) u_\Omega^\circ(\theta)) \, dx.$$

In order to transform the second term, we introduce the adjoint state p_Ω , the unique solution to (4.24), which reads under variational form:

$$(4.26) \quad \forall v \in H_0^1(\Omega), \quad \int_{\Omega} \nabla p_\Omega \cdot \nabla v \, dx = - \int_{\Omega} j'(u_\Omega) v \, dx$$

Since $u_\Omega^\circ(\theta)$ belongs to $H_0^1(\Omega)$, the latter can be used as a test function in the above formulation, which yields:

$$(4.27) \quad \begin{aligned} \int_{\Omega} j'(u_\Omega) u_\Omega^\circ(\theta) \, dx &= - \int_{\Omega} \nabla p_\Omega \cdot \nabla (u_\Omega^\circ(\theta)) \, dx \\ &= \int_{\Omega} ((\operatorname{div} \theta) \mathbf{I} - (\nabla \theta + \nabla \theta^T)) \nabla u_\Omega \cdot \nabla p_\Omega \, dx \\ &\quad - \int_{\Omega} \operatorname{div}(f\theta) p_\Omega \, dx, \end{aligned}$$

where we have used the variational formulation (4.15) of $u_\Omega^\circ(\theta)$ with p_Ω as test function to pass from the first line to the second one. Combining (4.25) with (4.27) yields the desired formula (4.23). \square

At this point, one may wonder how the defining formula (4.24) for the adjoint state p_Ω was coined. Actually, the variational formulation (4.26) for p_Ω is simply and exactly what is needed for the first line of (4.27) to hold.

This adjoint state “trick” is perhaps clearer when it is cast into an abstract context. Let us write the variational formulation (4.15) for the Lagrangian derivative $u_\Omega^\circ(\theta)$ under the form:

$$(4.28) \quad \forall w \in V, \quad a(u_\Omega^\circ(\theta), w) = -b(\theta)(w),$$

where V is a Hilbert space, $a : V \times V \rightarrow \mathbb{R}$ is a coercive bilinear form, and the coefficients of the linear form $b(\theta) : V \rightarrow \mathbb{R}$ depend on θ . The adjoint state method allows to rewrite a quantity of interest of the form $\ell(u_\Omega^\circ(\theta))$, where $\ell : V \rightarrow \mathbb{R}$ is a linear mapping; in the proof of Proposition 4.5, ℓ stands for the second term in the right-hand side of (4.25). More precisely, an explicit expression of the latter with respect to θ

(i.e. where the intricate dependence on θ via $u_\Omega^\circ(\theta)$ is eliminated) is obtained by introducing an adjoint state $p_\Omega \in V$ satisfying:

$$(4.29) \quad \forall v \in V, \quad a(v, p_\Omega) = -\ell(v).$$

Combining (4.28) and (4.29) then yields:

$$\begin{aligned} \ell(u_\Omega^\circ(\theta)) &= -a(u_\Omega^\circ(\theta), p_\Omega) \\ &= b(\theta)(p_\Omega), \end{aligned}$$

which corresponds exactly to (4.27) in the proof of Proposition 4.5, and results in the desired explicit expression of $\ell(u_\Omega^\circ(\theta))$.

Remark 4.6. *In the case of the function $J(\Omega)$ in (4.11), the adjoint state p_Ω , characterized by (4.24), has an interesting physical interpretation: it is the field generated by the artificial source $-j'(u_\Omega)$, taking larger values (in modulus) at points $x \in \Omega$ where the integrand $j(u_\Omega)$ of $J(\Omega)$ is more sensitive to variations of the state u_Ω .*

4.5.3. Surface form of the shape derivative

The explicit expression (4.23) of the shape derivative $J'(\Omega)(\theta)$ with respect to θ makes it possible to extract descent directions for $J(\Omega)$, although this task is not completely trivial; see Section 5.2.

Assuming more regularity from u_Ω and p_Ω , it is actually possible to ease this task further, by transforming (4.23) into a surface integral, of the convenient form (4.8). Classically, this operation demands a little more regularity on u_Ω and p_Ω , which is usually predicted by elliptic regularity theory (see Remark 4.4).

Theorem 4.6. *Assume that u_Ω and p_Ω are in $H^2(\Omega)$. Then,*

$$(4.30) \quad \forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad J'(\Omega)(\theta) = \int_{\partial\Omega} j(u_\Omega) \theta \cdot n \, ds - \int_{\partial\Omega} \frac{\partial u_\Omega}{\partial n} \frac{\partial p_\Omega}{\partial n} \theta \cdot n \, ds - \int_{\partial\Omega} f p_\Omega \theta \cdot n \, ds.$$

Proof. The strategy consists in using a series of integration by parts to rewrite (4.23) under the form:

$$(4.31) \quad J'(\Omega)(\theta) = \int_{\Omega} s_\Omega \cdot \theta \, dx + \int_{\partial\Omega} t_\Omega \cdot \theta_T \, ds + \int_{\partial\Omega} v_\Omega \theta \cdot n \, ds,$$

for some vector fields $s_\Omega : \Omega \rightarrow \mathbb{R}^d$, $t_\Omega : \partial\Omega \rightarrow \mathbb{R}^d$, and scalar field $v_\Omega : \partial\Omega \rightarrow \mathbb{R}$. We recall from Definition 4.2 that $\theta_T = \theta - (\theta \cdot n)n$ is the tangential component of θ . This expression (4.31) is a convenient intermediate to work towards the desired surface structure (4.8) for $J'(\Omega)(\theta)$, and it allows to identify which terms should cancel. Indeed, (4.31) is of the form (4.8) if and only if $s_\Omega = 0$ and $t_\Omega = 0$, as follows from comparing both expressions using at first arbitrary deformations θ with compact support inside Ω , then deformations with vanishing normal component and arbitrary tangential trace.

In order to work towards an expression of the form (4.31) from the volume form (4.23), we rely on the following integration by parts formulas (which are straightforward consequences of the usual Green's identity), valid for deformations $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, functions $w \in H^1(\Omega)$, and vector fields $a, b \in H^1(\Omega)^d$:

$$(4.32) \quad \int_{\Omega} \operatorname{div} \theta w \, dx = \int_{\partial\Omega} w \theta \cdot n \, ds - \int_{\Omega} \theta \cdot \nabla w \, dx;$$

$$(4.33) \quad \int_{\Omega} \nabla \theta a \cdot b \, dx = \int_{\partial\Omega} (\theta \cdot b)(a \cdot n) \, ds - \int_{\Omega} (\operatorname{div} a)(\theta \cdot b) \, dx - \int_{\Omega} (\nabla b) a \cdot \theta \, dx.$$

Based on the assumption $\nabla u_\Omega, \nabla p_\Omega \in H^1(\Omega)^d$, we obtain from (4.23) that:

$$(4.34) \quad \begin{aligned} J'(\Omega)(\theta) &= \int_{\partial\Omega} j(u_\Omega) \theta \cdot n \, ds - \int_{\Omega} j'(u_\Omega) \nabla u_\Omega \cdot \theta \, dx \\ &\quad + \int_{\partial\Omega} \left((\nabla u_\Omega \cdot \nabla p_\Omega) \theta \cdot n - (\nabla u_\Omega \cdot \theta) \frac{\partial p_\Omega}{\partial n} - (\nabla p_\Omega \cdot \theta) \frac{\partial u_\Omega}{\partial n} \right) ds \\ &\quad - \int_{\Omega} (\nabla(\nabla u_\Omega \cdot \nabla p_\Omega) \cdot \theta - \nabla^2 p_\Omega \nabla u_\Omega \cdot \theta - \nabla^2 u_\Omega \nabla p_\Omega \cdot \theta) \, dx \\ &\quad + \int_{\Omega} (\Delta u_\Omega (\theta \cdot \nabla p_\Omega) + \Delta p_\Omega (\theta \cdot \nabla u_\Omega)) \, dx \\ &\quad - \int_{\partial\Omega} f p_\Omega \theta \cdot n \, ds + \int_{\Omega} f \nabla p_\Omega \cdot \theta \, dx. \end{aligned}$$

According to the previous discussion, we verify that all the integrals supported on Ω in the above formula cancel, that is:

$$\begin{aligned} - \int_{\Omega} j'(u_{\Omega}) \nabla u_{\Omega} \cdot \theta \, dx - \int_{\Omega} (\nabla(\nabla u_{\Omega} \cdot \nabla p_{\Omega}) \cdot \theta - \nabla^2 p_{\Omega} \nabla u_{\Omega} \cdot \theta - \nabla^2 u_{\Omega} \nabla p_{\Omega} \cdot \theta) \, dx \\ + \int_{\Omega} (\Delta u_{\Omega} (\theta \cdot \nabla p_{\Omega}) + \Delta p_{\Omega} (\theta \cdot \nabla u_{\Omega})) \, dx + \int_{\Omega} f \nabla p_{\Omega} \cdot \theta \, dx = 0, \end{aligned}$$

as follows from the facts that:

$$-\Delta u_{\Omega} = f, \quad -\Delta p_{\Omega} = -j'(u_{\Omega}),$$

and the easy identity:

$$\nabla(\nabla u_{\Omega} \cdot \nabla p_{\Omega}) = \nabla^2 u_{\Omega} \nabla p_{\Omega} + \nabla^2 p_{\Omega} \nabla u_{\Omega}.$$

In addition, since u_{Ω} and p_{Ω} vanish on $\partial\Omega$ (thus so do their tangential derivatives), the second line in the right-hand side of (4.34) rewrites:

$$\int_{\partial\Omega} \left((\nabla u_{\Omega} \cdot \nabla p_{\Omega}) \theta \cdot n - (\nabla u_{\Omega} \cdot \theta) \frac{\partial p_{\Omega}}{\partial n} - (\nabla p_{\Omega} \cdot \theta) \frac{\partial u_{\Omega}}{\partial n} \right) \, ds = - \int_{\partial\Omega} \frac{\partial u_{\Omega}}{\partial n} \frac{\partial p_{\Omega}}{\partial n} \theta \cdot n \, ds,$$

so that we end up with the desired expression (4.30). \square

To conclude, we have obtained two different expressions of the shape derivative $J'(\Omega)(\theta)$, which are both explicit with respect to θ in that its calculation does not require the resolution of a partial differential equation involving θ . The *volume*, or *distributed expression* (4.23) of the shape derivative has the form of an integral over the whole shape Ω , featuring the state u_{Ω} , the adjoint p_{Ω} and θ . The *surface expression* (4.30) can be derived from the volume form when u_{Ω} and p_{Ω} are more regular. It features an integral on the boundary $\partial\Omega$, depending on u_{Ω} , p_{Ω} , and the normal component $\theta \cdot n$.

It is quite general that shape derivatives enjoy these two different, albeit equivalent structures. Which one is the most useful depends very much on the purpose. On the one hand, the surface form obviously complies with the “nice” structure (4.8), which allows for a simple identification of a descent direction for $J(\Omega)$ (see Section 4.2.3). On the other hand, the volume form requires minimum regularity from u_{Ω} and p_{Ω} ; besides, it is reported to show better numerical stability, and it is sometimes preferred for theoretical analyses; see [119] about this point.

Remark 4.7. *As we have mentioned, the methodology used to derive Theorem 4.6 is quite general: in particular it applies in an analogous way when the state equation (4.10) features different boundary conditions (e.g. homogeneous Neumann boundary conditions), or when it is replaced by a different physical equation (e.g. by the linearized elasticity system (2.1)).*

4.6. Shape derivative of PDE constrained shape functionals: C ea’s formal method

In this section, we present the method of C ea, which was introduced in [59] as a fast and formal calculation technique of the derivative of a wide variety of shape functionals. In particular, it alleviates the tedious identification of the Lagrangian and Eulerian derivatives of the state u_{Ω} .

In a nutshell, C ea’s method advocates to see the quantity $J(\Omega)$ as the saddle point of a Lagrangian function $(u, p) \mapsto \mathcal{L}(\Omega, u, p)$. The $(\Omega$ -dependent) arguments u, p ensuring that this saddle point property holds are exactly the solutions u_{Ω} and p_{Ω} to the state and adjoint equations. The calculation of the shape derivative of $J(\Omega)$ is then greatly simplified by the fact that it amounts to the derivative of $\Omega \mapsto \mathcal{L}(\Omega, u, p)$ evaluated at $u = u_{\Omega}$ and $p = p_{\Omega}$, i.e. only the (simple) partial derivative of \mathcal{L} with respect to its *explicit* dependence on Ω is needed; see Figure 10 for an illustration.

This method also offers an interesting interpretation of the adjoint state. Consider indeed the unconstrained shape optimization problem (2.9) involving the shape functional $J(\Omega)$ in (4.11): this problem can be rewritten in such a way that the defining partial differential equation (4.10) for u_{Ω} is a constraint:

$$(4.35) \quad \min_{(\Omega, u)} \int_{\Omega} j(u) \, dx, \quad \text{s.t.} \quad \begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

As we shall see below, the adjoint state p_{Ω} then turns out to be the Lagrange multiplier for the constraint that u should satisfy (4.10).

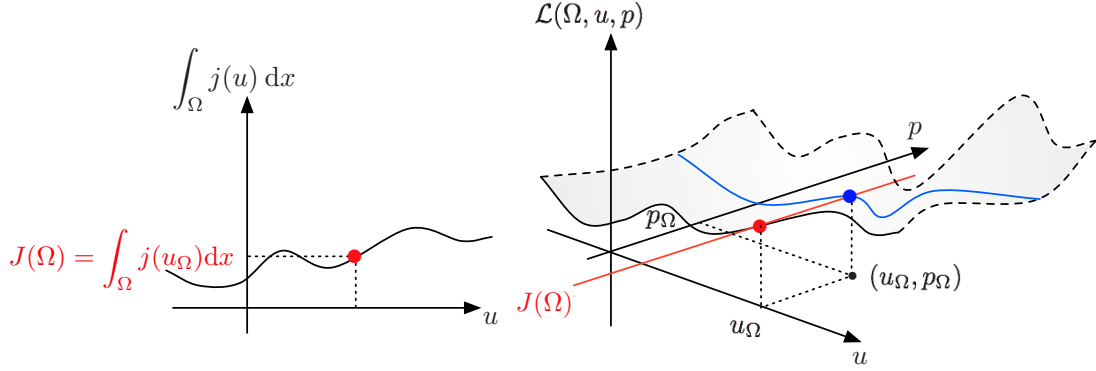


FIGURE 10. C ea's method consists in expressing the value $J(\Omega)$ as the saddle point of a Lagrangian $\mathcal{L}(\Omega, u, p)$.

Let us insist that the method of C ea is formal, i.e., not rigorous, because it assumes that the state $\Omega \mapsto u_\Omega$ has a Eulerian derivative. Nevertheless, if this assumption is indeed satisfied, then the method becomes fully rigorous, see the discussion below in Subsection 4.6.3.

To simplify our discussion in this section, we systematically assume that the considered shapes are ‘‘smooth enough’’, which we encode by introducing a set \mathcal{U}_{ad} of smooth admissible shapes. The deformations θ featured in the method of Hadamard accordingly belong to a set Θ_{ad} of ‘‘smooth’’ vector fields; see Section 4.1.

4.6.1. The case of Neumann boundary conditions

We start by presenting C ea's method in the simpler context where the state u_Ω satisfies homogeneous Neumann boundary conditions. More precisely, we consider again the calculation of the shape derivative of the functional:

$$(4.36) \quad J(\Omega) = \int_{\Omega} j(u_\Omega) \, dx,$$

where $j : \mathbb{R} \rightarrow \mathbb{R}$ is a smooth function satisfying the growth conditions (4.12), and u_Ω is the unique solution in $H^1(\Omega)$ to the problem:

$$(4.37) \quad \begin{cases} -\Delta u_\Omega + u_\Omega = f & \text{in } \Omega, \\ \frac{\partial u_\Omega}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases}$$

Note that this Laplace equation brings into play a 0th order term, whose sole purpose is to make the state problem well-posed. This feature does not change anything to our theoretical discussion, and this term will be dropped in the subsequent sections, when Dirichlet boundary conditions are considered. The variational formulation for (4.37) reads:

$$(4.38) \quad \forall v \in H^1(\Omega), \quad \int_{\Omega} (\nabla u_\Omega \cdot \nabla v + u_\Omega v) \, dx = \int_{\Omega} f v \, dx.$$

Our aim is to give a formal proof of the following result, whose rigorous counterpart can be achieved along the lines of Section 4.4

Theorem 4.7. *The functional $J(\Omega)$ defined in (4.36) is shape differentiable at any shape $\Omega \in \mathcal{U}_{\text{ad}}$, and its shape derivative reads:*

$$\forall \theta \in \Theta_{\text{ad}}, \quad J'(\Omega)(\theta) = \int_{\partial\Omega} \left(j(u_\Omega) + \nabla u_\Omega \cdot \nabla p_\Omega + u_\Omega p_\Omega - f p_\Omega \right) \theta \cdot n \, ds,$$

where the adjoint state p_Ω is the unique solution in $H^1(\Omega)$ to the equation:

$$(4.39) \quad \begin{cases} -\Delta p_\Omega + p_\Omega = -j'(u_\Omega) & \text{in } \Omega, \\ \frac{\partial p_\Omega}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases}$$

Formal proof. Let us introduce the *Lagrangian* functional:

$$\mathcal{L} : \mathcal{U}_{\text{ad}} \times H^1(\mathbb{R}^d) \times H^1(\mathbb{R}^d) \rightarrow \mathbb{R}$$

defined by:

$$\mathcal{L}(\Omega, u, p) = \int_{\Omega} j(u) \, dx + \int_{\Omega} (\nabla u \cdot \nabla p + up) \, dx - \int_{\Omega} fp \, dx.$$

In brief, \mathcal{L} is obtained from the considered functional $J(\Omega)$ by enforcing the partial differential equation constraint that u should be the solution to (4.37) with a Lagrange multiplier p ; see the constrained optimization version (4.35) of the problem (2.9). In particular, it is clear from the variational formulation (4.38) that for any $p \in H^1(\mathbb{R}^d)$,

$$(4.40) \quad J(\Omega) = \mathcal{L}(\Omega, u_{\Omega}, p).$$

For a given shape $\Omega \in \mathcal{U}_{\text{ad}}$, we now search for all the critical points (u, p) of $\mathcal{L}(\Omega, \cdot, \cdot)$, i.e. those points at which $\frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p) = \frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p) = 0$.

- The partial derivative $\frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p)$ reads:

$$\forall \hat{p} \in H^1(\mathbb{R}^d), \quad \frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p)(\hat{p}) = \int_{\Omega} (\nabla u \cdot \nabla \hat{p} + u\hat{p}) \, dx - \int_{\Omega} f\hat{p} \, dx.$$

Imposing that this quantity vanish for all $\hat{p} \in \mathcal{C}_c^{\infty}(\Omega)$, in particular, and integrating by parts, we obtain that

$$-\Delta u + u = f \text{ on } \Omega.$$

Considering now any $\hat{p} \in H^1(\mathbb{R}^d)$, integrating by parts yields the alternative expression:

$$\frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p)(\hat{p}) = \int_{\Omega} \frac{\partial u}{\partial n} \hat{p} \, ds.$$

Imposing that this derivative vanish for all $\hat{p} \in H^1(\mathbb{R}^d)$ (with arbitrary trace on $\partial\Omega$), we infer that

$$\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega.$$

Hence, u is uniquely characterized as $u = u_{\Omega}$, the solution to (4.37).

- The partial derivative of \mathcal{L} with respect to u reads:

$$\forall \hat{u} \in H^1(\mathbb{R}^d), \quad \frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p)(\hat{u}) = \int_{\Omega} j'(u)\hat{u} \, dx + \int_{\Omega} (\nabla \hat{u} \cdot \nabla p + \hat{u}p) \, dx.$$

Imposing that it vanish for all $\hat{u} \in \mathcal{C}_c^{\infty}(\Omega)$ and integrating by parts, we obtain:

$$-\Delta p + p = -j'(u).$$

Now considering arbitrary variations $\hat{u} \in H^1(\mathbb{R}^d)$, another integration by parts yields:

$$\frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p)(\hat{u}) = \int_{\partial\Omega} \frac{\partial p}{\partial n} \hat{u} \, ds.$$

Since the latter quantity should vanish for $\hat{u} \in H^1(\mathbb{R}^d)$ with arbitrary trace on $\partial\Omega$, it follows:

$$\frac{\partial p}{\partial n} = 0 \text{ on } \partial\Omega.$$

Finally, since $u = u_{\Omega}$, we identify p as p_{Ω} , the unique solution to the adjoint equation (4.39).

To conclude about the shape derivative of $J(\Omega)$, we return to (4.40). Fixing an arbitrary $p \in H^1(\mathbb{R}^d)$, taking derivatives with respect to the domain in this identity and using the chain rule produces:

$$(4.41) \quad \forall \theta \in \Theta_{\text{ad}}, \quad J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_{\Omega}, p)(\theta) + \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_{\Omega}, p)(u'_{\Omega}(\theta)),$$

where $u'_{\Omega}(\theta)$ is the Eulerian derivative of the mapping $\Omega \mapsto u_{\Omega}$. Since this holds for arbitrary $p \in H^1(\mathbb{R}^d)$, we may now select $p = p_{\Omega}$, a very convenient choice since the partial derivative $\frac{\partial \mathcal{L}}{\partial u}(\Omega, u_{\Omega}, p_{\Omega})$ then vanishes. Hence,

$$\forall \theta \in \Theta_{\text{ad}}, \quad J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_{\Omega}, p_{\Omega})(\theta).$$

The latter quantity is simply the derivative of the mapping

$$\Omega \mapsto \int_{\Omega} j(u) \, dx + \int_{\Omega} (\nabla u \cdot \nabla p + up) \, dx - \int_{\Omega} fp \, dx$$

with respect to the explicit dependence with respect to the domain, taken at $u = u_{\Omega}$ and $p = p_{\Omega}$. A simple calculation based on Theorem 4.2 yields the desired result. \square

4.6.2. The case of Dirichlet boundary conditions

We return to the problem introduced at the beginning of Section 4.3, and we calculate the shape derivative of the functional $J(\Omega)$ defined in (4.11), featuring the solution u_{Ω} to the Laplace equation (4.10) with homogeneous Dirichlet boundary conditions. More precisely, we give a formal proof of Theorem 4.6, whose statement is reproduced below for the reader's convenience.

Theorem 4.8. *The functional $J(\Omega)$ defined in (4.11) is shape differentiable at any shape $\Omega \in \mathcal{U}_{\text{ad}}$, and its shape derivative reads:*

$$\forall \theta \in \Theta_{\text{ad}}, \quad J'(\Omega)(\theta) = \int_{\partial\Omega} \left(j(u_{\Omega}) - \frac{\partial u_{\Omega}}{\partial n} \frac{\partial p_{\Omega}}{\partial n} \right) \theta \cdot n \, ds,$$

where the adjoint state p_{Ω} is the unique solution in $H_0^1(\Omega)$ to the equation:

$$(4.42) \quad \begin{cases} -\Delta p_{\Omega} = -j'(u_{\Omega}) & \text{in } \Omega, \\ p_{\Omega} = 0 & \text{on } \partial\Omega. \end{cases}$$

Formal proof. The present case, where the state equation (4.10) for u_{Ω} features Dirichlet boundary conditions, is a little more subtle to handle with C ea's method than that in the previous section, where Neumann boundary conditions were considered. Indeed, using the naive definition of the Lagrangian, inspired from that considered in the proof of Theorem 4.7,

$$(4.43) \quad \mathcal{L}(\Omega, u, p) = \int_{\Omega} j(u) \, dx + \int_{\Omega} (-\Delta u - f)p \, dx$$

and conducting calculations in an analogous way as in there yield an erroneous formula for the shape derivative of $J(\Omega)$! This is because the use of the Lagrangian (4.43) imposes that the variables u and p belong to the space $H_0^1(\Omega)$, so that the three variables Ω , u and p of \mathcal{L} are not independent from one another.

To remedy this issue, we rather consider the Lagrangian:

$$\mathcal{L} : \mathcal{U}_{\text{ad}} \times H^1(\mathbb{R}^d) \times H^1(\mathbb{R}^d) \times H^1(\mathbb{R}^d) \rightarrow \mathbb{R}$$

defined by:

$$\mathcal{L}(\Omega, u, p, \lambda) = \int_{\Omega} j(u) \, dx + \int_{\Omega} (-\Delta u - f)p \, dx + \int_{\partial\Omega} \lambda u \, ds.$$

Note that the additional variable λ plays the role of a Lagrange multiplier for the Dirichlet boundary condition satisfied by u_{Ω} . Again, it holds, for arbitrary $p \in H^1(\mathbb{R}^d)$, $\lambda \in H^1(\mathbb{R}^d)$,

$$(4.44) \quad J(\Omega) = \mathcal{L}(\Omega, u_{\Omega}, p, \lambda).$$

Now, for a given shape $\Omega \in \mathcal{U}_{\text{ad}}$, we search for the critical points (u, p, λ) of $\mathcal{L}(\Omega, \cdot, \cdot, \cdot)$:

- The cancellation of the partial derivative $\frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p, \lambda)$ reads:

$$\forall \hat{p} \in H^1(\mathbb{R}^d), \quad \int_{\Omega} -\Delta u \hat{p} \, dx - \int_{\Omega} f \hat{p} \, dx = 0.$$

In particular, imposing that this hold for $\hat{p} \in \mathcal{C}_c^{\infty}(\Omega)$ yields:

$$(4.45) \quad -\Delta u = f \text{ in } \Omega.$$

- That the partial derivative $\frac{\partial \mathcal{L}}{\partial \lambda}(\Omega, u, p, \lambda)$ vanish rewrites:

$$\forall \hat{\lambda} \in H^1(\mathbb{R}^d), \quad \int_{\partial\Omega} \hat{\lambda} u \, ds = 0.$$

This readily implies that:

$$(4.46) \quad u = 0 \text{ on } \partial\Omega.$$

Then, (4.45) and (4.46) allow to identify $u = u_\Omega$, the solution to (4.10).

- Finally, the cancellation of the partial derivative $\frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p, \lambda)$ reads:

$$\forall \widehat{u} \in H^1(\mathbb{R}^d), \quad \int_{\Omega} j'(u) \widehat{u} \, dx + \int_{\Omega} -\Delta \widehat{u} p \, dx + \int_{\partial\Omega} \lambda \widehat{u} \, ds = 0.$$

Integrating by parts twice in the above equation, we obtain that, for arbitrary $\widehat{u} \in H^1(\mathbb{R}^d)$:

$$\int_{\Omega} j'(u) \widehat{u} \, dx - \int_{\partial\Omega} \frac{\partial \widehat{u}}{\partial n} p \, ds + \int_{\partial\Omega} \frac{\partial p}{\partial n} \widehat{u} \, ds + \int_{\Omega} -\Delta p \widehat{u} \, dx + \int_{\partial\Omega} \lambda \widehat{u} \, ds = 0.$$

Using at first arbitrary functions $\widehat{u} \in \mathcal{C}_c^\infty(\Omega)$ in the above identity yields:

$$(4.47) \quad -\Delta p = -j'(u) \text{ in } \Omega.$$

Then, using functions $\widehat{u} \in \mathcal{C}^\infty(\overline{\Omega})$ with arbitrary normal derivative $\frac{\partial \widehat{u}}{\partial n}$ on $\partial\Omega$ and vanishing trace ($\widehat{u} = 0$), it follows:

$$(4.48) \quad p = 0 \text{ on } \partial\Omega.$$

Combining (4.47) and (4.48) with the fact that we have already identified u as the solution u_Ω to (4.10), we see that $p = p_\Omega$, the unique solution to the adjoint system (4.42). Finally, taking $\widehat{u} \in \mathcal{C}^\infty(\overline{\Omega})$ with arbitrary trace on $\partial\Omega$ and vanishing normal derivative, λ equals:

$$(4.49) \quad \lambda = \lambda_\Omega := -\frac{\partial p_\Omega}{\partial n} \text{ on } \partial\Omega.$$

To conclude, let us return to (4.46). Taking derivatives with respect to the domain and using the chain rule, it holds, for arbitrary $p \in H^1(\mathbb{R}^d)$ and $\lambda \in H^1(\mathbb{R}^d)$,

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p, \lambda)(\theta) + \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p, \lambda)(u'_\Omega(\theta)).$$

Then, choosing $p = p_\Omega$ and $\lambda = \lambda_\Omega$, the last partial derivative vanishes, so that we end up with:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, \lambda_\Omega)(\theta),$$

a derivative with respect to the domain which can be calculated by means of Theorems 4.2 and 4.3. A simple calculation yields indeed:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} j(u_\Omega) \theta \cdot n \, ds + \int_{\partial\Omega} (-\Delta u_\Omega - f) p_\Omega \theta \cdot n \, ds + \int_{\partial\Omega} \left(\frac{\partial}{\partial n} + \kappa \right) (\lambda_\Omega u_\Omega) \theta \cdot n \, ds.$$

Using the state equation (4.10), the second integral in the above right-hand side vanishes. Since $u_\Omega = 0$ on $\partial\Omega$, the last integral simply rewrites:

$$\int_{\partial\Omega} \left(\frac{\partial}{\partial n} + \kappa \right) (\lambda_\Omega u_\Omega) \theta \cdot n \, ds = \int_{\partial\Omega} \lambda_\Omega \frac{\partial u_\Omega}{\partial n} \theta \cdot n \, ds,$$

and the desired result follows from the definition (4.49) of λ_Ω . \square

4.6.3. Final comments about C ea's method

As exemplified by the previous sections, if handled with care, C ea's method conveniently yields the correct formula for the shape derivative of a function of the domain without requiring the tedious calculation of the derivative of the mapping $\Omega \mapsto u_\Omega$.

Let us however stress once again that this method is *formal* in general, even though it can be made rigorous in certain cases, thanks to a theorem for differentiating a saddle point with respect to a parameter; see [82], Chap. 10.

The main requirement underlying this method is that the mapping $\Omega \mapsto u_\Omega$ should have an Eulerian derivative, as transpires from (4.41) in the proof of Theorem 4.7. This may fail to happen, for very clear physical reasons, for instance when the function u_Ω is not regular enough (in the context of Definition 4.3, if ∇u_Ω does not belong to $H^1(\Omega)$). Two such situations are:

- The case where the optimized shape Ω is one phase inside a composite medium; the solution u_Ω satisfies transmission conditions at the interface $\partial\Omega$ which prevent it from being globally regular; see [149] and Section 4.7 below for a detailed discussion.
- The case where the state equation for u_Ω features a decomposition of $\partial\Omega$ into several regions, bearing different types of boundary conditions (e.g. homogeneous Dirichlet or Neumann conditions, as in (2.1)). This happens frequently in applications, as we shall see for instance in the numerical examples of Section 7.4, but often, the separation between these regions is fixed during the optimization. When such is not the case (a situation that we rule out in the following), a particular care is in order about the lack of regularity of u_Ω near these separations; see [110] and [77, 100] in the shape optimization context.

One of the dangers of C ea’s method is that it yields an erroneous result if no particular care is paid to these considerations—a point that we illustrate in the next section.

4.7. Optimization of the shape of the interface between two different media

We eventually consider a physical model close to that in Section 3.1: inside a fixed bounded domain D , the optimized shape $\Omega \Subset D$ delimits two phases $\Omega_0 := \Omega$ and $\Omega_1 := D \setminus \overline{\Omega}$, filled with materials with different, positive conductivities α, β , respectively. Assuming Ω to be Lipschitz, we denote by n the unit normal vector to $\partial\Omega$ pointing outward Ω .

Let us consider the two-phase conductivity equation:

$$(4.50) \quad \begin{cases} -\operatorname{div}(\gamma_\Omega \nabla u_\Omega) = f & \text{in } D, \\ u_\Omega = 0 & \text{on } \partial D, \end{cases} \quad \text{where } \gamma_\Omega(x) = \begin{cases} \beta & \text{if } x \in \Omega, \\ \alpha & \text{if } x \in D \setminus \Omega, \end{cases}$$

which has a unique solution $u_\Omega \in H_0^1(D)$; see Figure 11 for an illustration.

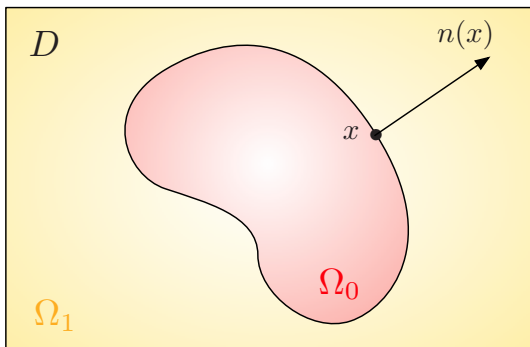


FIGURE 11. Setting of the two-phase conductivity equation considered in Section 4.7.

The formulation (4.50) is understood in the sense of distributions; in particular, it encompasses the *transmission conditions* of u_Ω across the interface $\partial\Omega$. Letting $u_\Omega^i \in H^1(\Omega_i)$ be the restriction of u_Ω to Ω_i , $i = 0, 1$, these read:

$$(4.51) \quad u_\Omega^0 = u_\Omega^1 \text{ in } H^{\frac{1}{2}}(\partial\Omega), \text{ and } \alpha \frac{\partial u_\Omega^0}{\partial n} = \beta \frac{\partial u_\Omega^1}{\partial n} \text{ in } H^{-\frac{1}{2}}(\partial\Omega).$$

Remark 4.8. The conditions (4.51) imply that u_Ω does not in general belong to $H^2(D)$, since the normal derivative $\frac{\partial u_\Omega}{\partial n}$ is not continuous across $\partial\Omega$. However, it can be proved by standard elliptic regularity theory that, provided Ω is of class C^2 , the restrictions u_Ω^0 and u_Ω^1 belong to $H^2(\Omega_0)$ and $H^2(\Omega_1)$ respectively; see also Remark 4.4 in connection with this point.

In this context, the considered shapes belong the admissible set

$$\mathcal{U}_{\text{ad}} := \{\Omega \Subset D \text{ is bounded and Lipschitz}\},$$

and the corresponding set of deformations Θ_{ad} reads:

$$\Theta_{\text{ad}} = \{\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \text{ s.t. } \theta = 0 \text{ on } \partial D\},$$

where θ is imposed to vanish on ∂D so that the boundary ∂D of the hold-all domain is fixed during the optimization. We aim to calculate the shape derivative of the functional:

$$(4.52) \quad J(\Omega) = \int_D j(u_\Omega) \, dx,$$

where $j : \mathbb{R} \rightarrow \mathbb{R}$ is smooth and satisfies the growth conditions (4.12).

4.7.1. Rigorous calculation of the shape derivative $J'(\Omega)(\theta)$

Our purpose in this section is to prove the following theorem:

Theorem 4.9. *Assume that $\Omega \Subset D$ is Lipschitz. Then, $J(\Omega)$ is shape differentiable at any admissible shape $\Omega \in \mathcal{U}_{\text{ad}}$ when deformations θ are considered in Θ_{ad} , and the shape derivative reads, in volume form:*

$$(4.53) \quad J'(\Omega)(\theta) = \int_D \left(f \nabla p_\Omega \cdot \theta - j'(u_\Omega) \nabla u_\Omega \cdot \theta \right) \, dx + \int_D \left((\text{div} \theta) \mathbf{I} - (\nabla \theta + \nabla \theta^T) \right) \gamma_\Omega \nabla u_\Omega \cdot \nabla p_\Omega \, dx,$$

where the adjoint state p_Ω is the unique solution in $H_0^1(D)$ to:

$$(4.54) \quad \begin{cases} -\text{div}(\gamma_\Omega \nabla p_\Omega) = -j'(u_\Omega) & \text{in } D, \\ p_\Omega = 0 & \text{on } \partial D. \end{cases}$$

If in addition Ω is of class \mathcal{C}^2 , $J'(\Omega)(\theta)$ has the surfacic expression:

$$(4.55) \quad J'(\Omega)(\theta) = (\alpha - \beta) \int_{\partial \Omega} \nabla_T u_\Omega \cdot \nabla_T p_\Omega \theta \cdot n \, ds - \left(\frac{1}{\alpha} - \frac{1}{\beta} \right) \int_{\partial \Omega} \left(\gamma_\Omega \frac{\partial u_\Omega}{\partial n} \right) \left(\gamma_\Omega \frac{\partial p_\Omega}{\partial n} \right) \theta \cdot n \, ds.$$

Remark 4.9. *Formula (4.55) is well-defined owing to the transmission conditions (4.51): the trace of u_Ω is well-defined on $\partial \Omega$ as that of either restriction u_Ω^0 or u_Ω^1 , and it is smooth, provided Ω is (see Remark 4.8). Likewise, the normal flux $\gamma_\Omega \frac{\partial u_\Omega}{\partial n}$ is unambiguously defined on $\partial \Omega$ as either $\alpha \frac{\partial u_\Omega^0}{\partial n}$ or $\beta \frac{\partial u_\Omega^1}{\partial n}$.*

Sketch of proof. We proceed along the trail of Sections 4.4 and 4.5.

Step 1: Existence and variational formulation of the Lagrangian derivative $u_\Omega^\circ(\theta)$: We start with the variational formulation for u_{Ω_θ} :

$$\forall v \in H_0^1(D), \quad \int_D \gamma_{\Omega_\theta} \nabla u_{\Omega_\theta} \cdot \nabla v \, dx = \int_D f v \, dx.$$

Using once again change of variables, as well as the fact that $\gamma_{\Omega_\theta} \circ (\text{Id} + \theta) = \gamma_\Omega$, and taking test functions of the form $v = w \circ (\text{Id} + \theta)^{-1}$, for $w \in H_0^1(D)$, we obtain a variational formulation for the transported mapping

$$\Theta_{\text{ad}} \ni \theta \mapsto \overline{u_\Omega}(\theta) := u_{\Omega_\theta} \circ (\text{Id} + \theta) \in H_0^1(D);$$

it reads: for any $w \in H_0^1(D)$,

$$(4.56) \quad \int_D \gamma_\Omega A(\theta) \nabla \overline{u_\Omega}(\theta) \cdot \nabla w \, dx = \int_D |\det(\mathbf{I} + \nabla \theta)| f \circ (\text{Id} + \theta) w \, dx,$$

where again $A(\theta) = |\det(\mathbf{I} + \nabla \theta)| (\mathbf{I} + \nabla \theta)^{-1} (\mathbf{I} + \nabla \theta)^{-T}$.

A similar argument as in the proof of Theorem 4.5, based on the implicit function theorem, reveals that the mapping $\theta \mapsto \overline{u_\Omega}(\theta)$ is Fréchet differentiable at $\theta = 0$. Taking derivatives in (4.56) then yields the following variational formulation for the Lagrangian derivative $u_\Omega^\circ(\theta)$:

$$(4.57) \quad \forall w \in H_0^1(D), \quad \int_D \gamma_\Omega \nabla u_\Omega^\circ(\theta) \cdot \nabla w \, dx = \int_D \text{div}(f \theta) w \, dx - \int_D \left((\text{div} \theta) \mathbf{I} - (\nabla \theta + \nabla \theta^T) \right) \gamma_\Omega \nabla u_\Omega \cdot \nabla w \, dx.$$

Step 2: Volume form of the shape derivative using the adjoint state: Changing variables in the definition of $J(\Omega_\theta)$ yields:

$$J(\Omega_\theta) = \int_D |\det(\mathbf{I} + \nabla \theta)| j'(\overline{u_\Omega}(\theta)) \, dx;$$

as a result of the first step, the mapping $\theta \mapsto J(\Omega_\theta)$ is Fréchet differentiable around 0 in Θ_{ad} , and:

$$J'(\Omega)(\theta) = \int_D ((\text{div}\theta)j(u_\Omega) + j'(u_\Omega)u_\Omega^\circ(\theta)) \, dx.$$

On the other hand, the variational formulation for the adjoint state p_Ω given by (4.54) reads:

$$(4.58) \quad \forall w \in H_0^1(D), \quad \int_D \gamma_\Omega \nabla p_\Omega \cdot \nabla w \, dx = - \int_D j'(u_\Omega)w \, dx.$$

Hence, taking $w = u_\Omega^\circ(\theta)$ as test function in (4.58), then taking $w = p_\Omega$ as test function in (4.57), we obtain successively:

$$\begin{aligned} J'(\Omega)(\theta) &= \int_D (\text{div}\theta)j(u_\Omega) \, dx - \int_D \gamma_\Omega \nabla p_\Omega \cdot \nabla(u_\Omega^\circ(\theta)) \, dx \\ &= \int_D (\text{div}\theta)j(u_\Omega) \, dx - \int_D \text{div}(f\theta)p_\Omega \, dx \\ &\quad + \int_D ((\text{div}\theta)\text{I} - (\nabla\theta + \nabla\theta^T)) \gamma_\Omega \nabla u_\Omega \cdot \nabla p_\Omega \, dx. \end{aligned}$$

Using integration by parts in the first two integrals in the right-hand side, together with the boundary condition $\theta = 0$ on ∂D , the desired volume shape derivative (4.53) follows.

Step 3: Integration by parts towards the surfacic form of the shape derivative: Using integration by parts inside the last integral in formula (4.53) we work toward an expression of $J'(\Omega)(\theta)$ of the form (4.31), which makes it is simple to identify which volume integrals should cancel. Since u_Ω and p_Ω are not in $H^2(D)$ (only their restrictions to Ω_0 and Ω_1 are; see Remark 4.8), we decompose the integrals in (4.53) on Ω_0 and Ω_1 :

$$(4.59) \quad J'(\Omega)(\theta) = \sum_{i=0}^1 \left(\int_{\Omega_i} (f \nabla p_\Omega^i \cdot \theta - j'(u_\Omega^i) \nabla u_\Omega^i \cdot \theta) \, dx + \int_{\Omega_i} ((\text{div}\theta)\text{I} - (\nabla\theta + \nabla\theta^T)) \gamma_\Omega \nabla u_\Omega^i \cdot \nabla p_\Omega^i \, dx \right).$$

Using the integration by parts formulas (4.32,4.33), we obtain as in the proof of Theorem 4.6:

$$(4.60) \quad \int_{\Omega_0} (f \nabla p_\Omega^0 \cdot \theta - j'(u_\Omega^0) \nabla u_\Omega^0 \cdot \theta) \, dx + \alpha \int_{\Omega_0} ((\text{div}\theta)\text{I} - (\nabla\theta + \nabla\theta^T)) \nabla u_\Omega^0 \cdot \nabla p_\Omega^0 \, dx = \\ \alpha \int_{\partial\Omega} \left((\nabla u_\Omega^0 \cdot \nabla p_\Omega^0) \theta \cdot n - (\nabla u_\Omega^0 \cdot \theta) \frac{\partial p_\Omega^0}{\partial n} - (\nabla p_\Omega^0 \cdot \theta) \frac{\partial u_\Omega^0}{\partial n} \right) \, ds,$$

whence, introducing the continuous quantities across the boundary, a straightforward calculation yields:

$$(4.61) \quad \int_{\Omega_0} (f \nabla p_\Omega^0 \cdot \theta - j'(u_\Omega^0) \nabla u_\Omega^0 \cdot \theta) \, dx + \alpha \int_{\Omega_0} ((\text{div}\theta)\text{I} - (\nabla\theta + \nabla\theta^T)) \nabla u_\Omega^0 \cdot \nabla p_\Omega^0 \, dx = \\ \int_{\partial\Omega} \left((\alpha \nabla_T u_\Omega^0 \cdot \nabla_T p_\Omega^0 - \frac{1}{\alpha} \left(\alpha \frac{\partial u_\Omega^0}{\partial n} \right) \left(\alpha \frac{\partial p_\Omega^0}{\partial n} \right) \right) \theta \cdot n \, ds \\ - \int_{\partial\Omega} \left((\nabla_T u_\Omega^0 \cdot \theta_T) \left(\alpha \frac{\partial p_\Omega^0}{\partial n} \right) + (\nabla_T p_\Omega^0 \cdot \theta_T) \left(\alpha \frac{\partial u_\Omega^0}{\partial n} \right) \right) \, ds.$$

Likewise, we obtain:

$$(4.62) \quad \int_{\Omega_1} (f \nabla p_\Omega^1 \cdot \theta - j'(u_\Omega^1) \nabla u_\Omega^1 \cdot \theta) \, dx + \alpha \int_{\Omega_1} ((\text{div}\theta)\text{I} - (\nabla\theta + \nabla\theta^T)) \nabla u_\Omega^1 \cdot \nabla p_\Omega^1 \, dx = \\ - \int_{\partial\Omega} \left((\beta \nabla_T u_\Omega^1 \cdot \nabla_T p_\Omega^1 - \frac{1}{\beta} \left(\beta \frac{\partial u_\Omega^1}{\partial n} \right) \left(\beta \frac{\partial p_\Omega^1}{\partial n} \right) \right) \theta \cdot n \, ds \\ + \int_{\partial\Omega} \left((\nabla_T u_\Omega^1 \cdot \theta_T) \left(\beta \frac{\partial p_\Omega^1}{\partial n} \right) + (\nabla_T p_\Omega^1 \cdot \theta_T) \left(\beta \frac{\partial u_\Omega^1}{\partial n} \right) \right) \, ds.$$

That the signs in front of the integrals in (4.61) and (4.62) differ is a consequence of the fact that n is pointing outward Ω_0 .

Combining (4.59), (4.61) and (4.62), we finally end up with the desired surface expression (4.55). \square

4.7.2. The use of C ea's method

We now describe in this section how C ea's method may be adapted to calculate the surface form (4.55) of the shape derivative of the function (4.52) in the present two-phase context, following the article [149]. We point out that applying C ea's method without caution in this situation yields the wrong result. The main reason is that the Eulerian derivative of u_Ω fails to exist on D in this case, because u_Ω does not belong to $H^2(D)$; see Remark 4.8 and Section 4.6.3.

We introduce the Lagrangian $\mathcal{L} : \mathcal{U}_{\text{ad}} \times (H_0^1(D))^6 \rightarrow \mathbb{R}$ defined by:

$$(4.63) \quad \mathcal{L}(\Omega, u_0, u_1, p_0, p_1, \lambda, \mu) = \int_{\Omega} j(u_0) \, dx + \int_{D \setminus \bar{\Omega}} j(u_1) \, dx + \int_{\Omega} (-\alpha \Delta u_0 - f) p_0 \, dx \\ + \int_{D \setminus \bar{\Omega}} (-\beta \Delta u_1 - f) p_1 \, dx + \int_{\partial\Omega} \lambda (u_1 - u_0) \, ds + \int_{\partial\Omega} \left(\beta \frac{\partial u_1}{\partial n} - \alpha \frac{\partial u_0}{\partial n} \right) \mu \, ds.$$

Note that in constructing \mathcal{L} , we have imposed that the function u should satisfy the state equation (4.50) on both subdomains Ω_0, Ω_1 by using two different Lagrange multipliers p_0, p_1 , and we have imposed the two transmission conditions (4.51) at the interface $\partial\Omega$ with two additional Lagrange multipliers λ and μ . As usual, we observe that:

$$(4.64) \quad \forall p_0, p_1, \lambda, \mu \in H^1(D), \quad J(\Omega) = \mathcal{L}(\Omega, u_\Omega^0, u_\Omega^1, p_0, p_1, \lambda, \mu).$$

For a given shape $\Omega \in \mathcal{U}_{\text{ad}}$, we now search for the saddle points $(u_0, u_1, p_0, p_1, \lambda, \mu)$ of the functional $\mathcal{L}(\Omega, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot)$.

- Cancelling the partial derivatives $\frac{\partial \mathcal{L}}{\partial p_0}, \frac{\partial \mathcal{L}}{\partial p_1}$ at $(u_0, u_1, p_0, p_1, \lambda, \mu)$, we see that u_0, u_1 satisfy:

$$-\alpha \Delta u_0 = f \text{ in } \Omega_0 \text{ and } -\beta \Delta u_1 = f \text{ in } \Omega_1.$$

Then cancelling the derivatives $\frac{\partial \mathcal{L}}{\partial \lambda}, \frac{\partial \mathcal{L}}{\partial \mu}$ yields the transmission conditions at the interface $\partial\Omega$:

$$u_0 = u_1 \text{ and } \alpha \frac{\partial u_0}{\partial n} = \beta \frac{\partial u_1}{\partial n}.$$

It follows from these facts that $u_0 = u_\Omega^0$ and $u_1 = u_\Omega^1$.

- Cancelling the partial derivatives $\frac{\partial \mathcal{L}}{\partial u_0}, \frac{\partial \mathcal{L}}{\partial u_1}$ at $(u_0, u_1, p_0, p_1, \lambda, \mu)$, we obtain, for arbitrary $\widehat{u}_0 \in H^1(\mathbb{R}^d)$,

$$(4.65) \quad \int_{\Omega} j'(u_0) \widehat{u}_0 \, dx - \alpha \int_{\Omega} \Delta \widehat{u}_0 p_0 \, dx - \int_{\partial\Omega} \lambda \widehat{u}_0 \, ds - \alpha \int_{\partial\Omega} \mu \frac{\partial \widehat{u}_0}{\partial n} \, ds = 0,$$

and, for arbitrary $\widehat{u}_1 \in H^1(\mathbb{R}^d)$,

$$(4.66) \quad \int_{D \setminus \bar{\Omega}} j'(u_1) \widehat{u}_1 \, dx - \beta \int_{D \setminus \bar{\Omega}} \Delta \widehat{u}_1 p_1 \, dx + \int_{\partial\Omega} \lambda \widehat{u}_1 \, ds + \beta \int_{\partial\Omega} \mu \frac{\partial \widehat{u}_1}{\partial n} \, ds = 0.$$

Integrating by parts twice in (4.65) yields:

$$\int_{\Omega} (-\alpha \Delta p_0 + j'(u_0)) \widehat{u}_0 \, dx + \int_{\partial\Omega} \left(\alpha \frac{\partial p_0}{\partial n} - \lambda \right) \widehat{u}_0 \, ds - \alpha \int_{\partial\Omega} \frac{\partial \widehat{u}_0}{\partial n} (\mu + p_0) \, ds = 0.$$

Now taking arbitrary $\widehat{u}_0 \in \mathcal{C}_c^\infty(\Omega)$, we obtain that:

$$-\alpha \Delta p_0 = -j'(u_0) \text{ in } \Omega.$$

Then, choosing $\widehat{u}_0 \in \mathcal{C}_c^\infty(D)$ with arbitrary trace on $\partial\Omega$ and null normal derivative $\frac{\partial \widehat{u}_0}{\partial n}$, then $\widehat{u}_0 \in \mathcal{C}_c^\infty(D)$ with null trace and arbitrary normal derivative, we obtain:

$$p_0 = -\mu \text{ and } \alpha \frac{\partial p_0}{\partial n} = \lambda \text{ on } \partial\Omega.$$

Similar calculations starting from (4.66) yield:

$$-\beta \Delta p_1 = -j'(u_1) \text{ in } D \setminus \bar{\Omega}, \quad p_1 = -\mu \text{ and } \beta \frac{\partial p_1}{\partial n} = \lambda \text{ on } \partial\Omega.$$

We infer that $p_0 = p_\Omega^0$ and $p_1 = p_\Omega^1$. In addition, the values of λ and μ are uniquely determined by:

$$(4.67) \quad \lambda = \lambda_\Omega := \gamma_\Omega \frac{\partial p_\Omega}{\partial n}, \text{ and } \mu = \mu_\Omega := -p_\Omega \text{ on } \partial\Omega.$$

To conclude, we return to (4.64), in which (formal) differentiation with respect to the domain yields, for arbitrary $p_0, p_1, \lambda, \mu \in H^1(\mathbb{R}^d)$,

$$\begin{aligned} J'(\Omega)(\theta) &= \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega^0, u_\Omega^1, p_0, p_1, \lambda, \mu)(\theta) \\ &\quad + \frac{\partial \mathcal{L}}{\partial u_0}(\Omega, u_\Omega^0, u_\Omega^1, p_0, p_1, \lambda, \mu)(u_\Omega^{0'}(\theta)) + \frac{\partial \mathcal{L}}{\partial u_1}(\Omega, u_\Omega^0, u_\Omega^1, p_0, p_1, \lambda, \mu)(u_\Omega^{1'}(\theta)), \end{aligned}$$

where $u_\Omega^{0'}(\theta)$ and $u_\Omega^{1'}(\theta)$ are the Eulerian derivatives of the respective mappings $\Omega \mapsto u_\Omega^0$ and $\Omega \mapsto u_\Omega^1$. Now taking $p_0 = p_\Omega^0$, $p_1 = p_\Omega^1$, $\lambda = \lambda_\Omega$ and $\mu = \mu_\Omega$, the above analysis shows that the last two partial derivatives in the above right-hand side vanish so that:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega^0, u_\Omega^1, p_\Omega^0, p_\Omega^1, \lambda_\Omega, \mu_\Omega)(\theta),$$

and we are left with the calculation of this derivative with respect to the explicit dependence of \mathcal{L} with respect to the domain.

Taking derivatives with respect to the domain in (4.63) using Theorems 4.2 and 4.3, and bringing into play the state equation (4.50) together with the continuity of u_Ω and p_Ω across $\partial\Omega$, the only possibly non vanishing terms are:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \left(\frac{\partial}{\partial n} + \kappa \right) (\lambda_\Omega (u_1 - u_0)) \theta \cdot n ds + \int_{\partial\Omega} \left(\frac{\partial}{\partial n} + \kappa \right) \left(\left(\beta \frac{\partial u_1}{\partial n} - \alpha \frac{\partial u_0}{\partial n} \right) \mu_\Omega \right) \theta \cdot n ds,$$

and so, using the values (4.67) of λ_Ω and μ_Ω together with the transmission conditions (4.51),

$$(4.68) \quad J'(\Omega)(\theta) = \int_{\partial\Omega} \gamma_\Omega \frac{\partial p_\Omega}{\partial n} \left(\frac{\partial u_1}{\partial n} - \frac{\partial u_0}{\partial n} \right) \theta \cdot n ds - \int_{\partial\Omega} p_\Omega \left(\beta \frac{\partial^2 u_1}{\partial n^2} - \alpha \frac{\partial^2 u_0}{\partial n^2} \right) \theta \cdot n ds.$$

In order to simplify further the second integral in the above right-hand side, we use the following decomposition on $\partial\Omega$:

$$\begin{aligned} f &= -\operatorname{div}(\alpha \nabla u_0) = -\operatorname{div}_T(\alpha \nabla u_0) - \alpha \frac{\partial^2 u_0}{\partial n^2} \\ &= -\operatorname{div}_T(\alpha \nabla_T u_0) - \operatorname{div}_T(\alpha \frac{\partial u_0}{\partial n} n) - \alpha \frac{\partial^2 u_0}{\partial n^2}, \end{aligned}$$

and likewise:

$$f = -\operatorname{div}_T(\beta \nabla_T u_1) - \operatorname{div}_T(\beta \frac{\partial u_1}{\partial n} n) - \beta \frac{\partial^2 u_1}{\partial n^2}.$$

Taking differences, we obtain:

$$(4.69) \quad \beta \frac{\partial^2 u_1}{\partial n^2} - \alpha \frac{\partial^2 u_0}{\partial n^2} = -\operatorname{div}_T(\beta \nabla_T u_1) + \operatorname{div}_T(\alpha \nabla_T u_0),$$

where we have used the fact that the vector fields $\alpha \frac{\partial u_0}{\partial n} n$ and $\beta \frac{\partial u_1}{\partial n} n$ coincide on $\partial\Omega$, and thus so do their tangential divergences. Inserting (4.69) into (4.68) and integrating by parts on $\partial\Omega$ using Proposition 4.3, we finally obtain:

$$J'(\Omega)(\theta) = \left(\frac{1}{\beta} - \frac{1}{\alpha} \right) \int_{\partial\Omega} \left(\gamma_\Omega \frac{\partial u_\Omega}{\partial n} \right) \left(\gamma_\Omega \frac{\partial p_\Omega}{\partial n} \right) \theta \cdot n ds - (\beta - \alpha) \int_{\partial\Omega} \nabla_T u_\Omega \cdot \nabla_T p_\Omega \theta \cdot n ds,$$

which is the desired expression.

Remark 4.10. *A slightly different approach for using C ea's method in the present context is proposed in [23]. The computation there is easier but it relies on the knowledge of the optimal values for the Lagrange multipliers λ and μ .*

5. SHAPE OPTIMIZATION USING HADAMARD’S METHOD: THEORETICAL FRAMEWORK

We now arrive at the central topic of this chapter, namely, the device of shape and topology optimization algorithms based on the sensitivity of quantities with respect to the domain provided by the method of Hadamard. We first sketch in Section 5.1 a naive shape gradient algorithm devoted to a model unconstrained optimization problem. This deliberately simple setting is used to set ideas and to highlight the main theoretical and numerical challenges posed by the implementation. Staying at the conceptual level, we then present in Section 5.2 the so-called *Hilbertian framework* for shape derivatives, a very convenient, multi-purpose paradigm. Finally, we discuss in Section 5.3 how constrained optimization problems can be addressed.

5.1. A basic shape gradient algorithm and some expected difficulties

Let us slip into the physical context of linear elastic structures. We again consider the unconstrained minimization problems (2.9), that is

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega),$$

where the objective function $J(\Omega)$ depends on the shape Ω via the solution u_Ω to the linearized elasticity system (2.1) (see for instance (2.10)).

We have seen in the previous Section 4 (see for instance Theorem 4.5.2) how to define and calculate the shape derivative of $J(\Omega)$; its expression involves in general u_Ω as well as an adjoint state p_Ω , which is the solution to another system of the form (2.1), with a different right-hand side.

These developments suggest the “shape gradient” Algorithm 1 for the numerical calculation of a (locally) optimal solution to (2.9). In Algorithm 1 we completely ignore any possible constraints, linked to the definition of the set of admissible shapes \mathcal{U}_{ad} .

Algorithm 1 Shape gradient algorithm for problem (2.9).

Initialization: Initial shape Ω^0 .

for $n = 0, \dots$, until convergence **do**

- (1) Calculate the solution u_{Ω^n} (resp. p_{Ω^n}) to the state (resp. adjoint) equation posed in Ω^n .
- (2) Calculate $\theta \mapsto J'(\Omega^n)(\theta)$ along the lines of Section 4.
- (3) Infer a descent direction θ^n for J from Ω^n , i.e. a vector field $\theta^n : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that $J'(\Omega^n)(\theta^n) < 0$.
- (4) Deform Ω^n along θ^n for a short descent step $\tau^n > 0$, so that the new shape

$$\Omega^{n+1} := (\text{Id} + \tau^n \theta^n)(\Omega^n)$$

is better than the previous one, i.e. $J(\Omega^{n+1}) < J(\Omega^n)$.

end for

return Ω^n

The implementation of this program poses difficulties of two natures.

- From the theoretical point of view, how can we identify a descent direction for $J(\Omega)$ from the knowledge of the shape derivative $J'(\Omega)$, as is required in Step 3? We have actually partially answered this question in Section 4.2.3, where we have seen that this task is quite simple when the structure (4.8) is observed by $J'(\Omega)(\theta)$. However, we shall see in Section 5.2.2 that much “better” descent directions can be obtained, thanks to the “Hilbertian extension-regularization procedure”, including when the structure (4.8) is not available. In addition, this technique allows to devise efficient methods for constrained optimization problems, as we discuss in Section 5.3.
- The question of how to represent shapes Ω and their deformations is crucial from the numerical point of view. On the one hand, this representation has to allow for the resolution of partial differential equations posed on Ω such as those defining u_Ω and p_Ω in Step 1 (for instance via the finite element method). On the other hand, it has to be robust enough to allow for large deformations of Ω in the update Step 4. We shall explore this numerical issue and provide several possible answers starting from the next Section 6.

5.2. The Hilbertian extension-regularization procedure

This section presents a very efficient and flexible framework to extract “nice” descent directions θ from the knowledge of the shape derivative $J'(\Omega)$. This method is part of the shape optimization “folklore” and it is difficult to trace back its inception. For example, it is already explained as a classical trick in section 9.5 of the textbook [138]. It is discussed in details in [53, 78, 156] for applications to shape optimization or inverse problems. The key idea is to play on the nature of the inner product whereby a *gradient* θ is inferred from the *differential* $J'(\Omega)$. To clarify this point, we start with a very elementary reminder (which may be omitted by the more expert reader) of classical background material about the notions of differential and gradient.

5.2.1. Foreword about differentials and gradients

Let H be a Hilbert space, with inner product $a(\cdot, \cdot)$ and associated norm $\|u\|_H := a(u, u)^{1/2}$. The *Fréchet derivative* (or *differential*) of a function $F : H \rightarrow \mathbb{R}$ at a point $u \in H$ is the linear, bounded mapping

$$F'(u) \in H^*, \quad v \mapsto F'(u)(v),$$

fulfilling the first-order expansion:

$$(5.1) \quad F(u + v) = F(u) + F'(u)(v) + o(v), \quad \text{where } \frac{|o(v)|}{\|v\|_H} \rightarrow 0 \text{ as } v \rightarrow 0.$$

As is well-known, the dual space H^* can be identified with H via the Riesz representation theorem; this allows to define the *gradient* of F at u as the unique element $g \in V$ such that:

$$\forall v \in H, \quad a(g, v) = F'(u)(v).$$

This gradient g depends on the inner product $a(\cdot, \cdot)$: if $\tilde{a}(\cdot, \cdot)$ is another inner product on H (equivalent to $a(\cdot, \cdot)$), the Riesz representation theorem also ensures that there is a unique element $\tilde{g} \in H$ such that:

$$\forall v \in H, \quad \tilde{a}(\tilde{g}, v) = F'(u)(v).$$

In general \tilde{g} differs from g , and depending on the application, it may be preferable to identify Fréchet derivatives with gradients according to \tilde{a} than to a .

Let now $(V, \|\cdot\|_V)$ be a Banach space, and $F : V \rightarrow \mathbb{R}$ be a differentiable function. The Fréchet derivative $F'(u)$ at $u \in V$ is still defined via the first-order expansion (5.1), but there is no associated notion of gradient since a Banach space is not, in general, equipped with an inner product. However, let $H \subset V$ be a Hilbert space with inner product $a(\cdot, \cdot)$ and norm $\|u\|_H = a(u, u)^{1/2}$, which is continuously embedded in V , that is:

$$\forall v \in H, \quad \|v\|_V \leq C\|v\|_H.$$

Then the derivative $V \ni v \mapsto F'(u)(v) \in \mathbb{R}$ naturally induces a bounded mapping on H . By the Riesz representation theorem, there exists $g \in H$ such that:

$$(5.2) \quad \forall v \in H, \quad a(g, v) = F'(u)(v);$$

that is, a gradient can be identified for the differential $F'(u)$ inside a Hilbert subspace H of V .

Let us finally consider the converse situation, which is of particular interest in applications. We still assume that $F : V \rightarrow \mathbb{R}$ is a differentiable function on the Banach space $(V, \|\cdot\|_V)$, but V is now continuously embedded in a larger Hilbert space H . We assume that the Fréchet derivative $v \mapsto F'(u)(v)$ has a continuous extension to H ; grossly speaking, we shall see in Section 5.2.2 that this is a kind of “regularity assumption” about $F : V \rightarrow \mathbb{R}$, resulting in that its derivative can be applied to a larger class of functions than those in V . The previous discussion applies verbatim in this new context, and one may then find a gradient $g \in H$ such that (5.2) holds. Note however that this identification of the derivative $F'(u)$ with a gradient $g \in H$, $g \notin V$ is quite formal since g cannot be interpreted as a variation of $u \in V$.

5.2.2. The change of inner products when using shape derivatives

Let us return to our purpose to use the first-order information contained in shape derivatives to solve the unconstrained minimization problem (2.9). As we have seen in Section 4.2.3, the shape derivatives of “most” (but not all) of the considered shape functionals in this chapter are of the form

$$(5.3) \quad \forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad J'(\Omega)(\theta) = \int_{\partial\Omega} v_\Omega \theta \cdot n \, ds = (v_\Omega n, \theta)_{L^2(\partial\Omega)^d},$$

where $v_\Omega : \partial\Omega \rightarrow \mathbb{R}$ is a scalar field which depends on the solution u_Ω to the state equation, and on the adjoint state p_Ω . Here, $(\cdot, \cdot)_{L^2(\partial\Omega)^d}$ stands for the usual inner product on $L^2(\partial\Omega)^d$. It is then tentative to take

$$(5.4) \quad \theta = -v_\Omega n \text{ on } \partial\Omega$$

when it comes to a descent direction for $J(\Omega)$ (Step 3 in Algorithm 1). Indeed, the Definition 4.1 of shape derivatives with this particular value of θ reads, for a small enough time step $\tau > 0$:

$$J(\Omega_{\tau\theta}) = J(\Omega) - \tau \int_{\partial\Omega} v_\Omega^2 \, ds < J(\Omega),$$

unless v_Ω vanishes identically on $\partial\Omega$, indicating that Ω is a stationary point of $J(\Omega)$. Using the language of Section 5.2.1, and omitting for a moment the fact that $L^2(\partial\Omega)^d$ is not a subset of $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, (5.4) corresponds to the gradient associated to the Fréchet derivative $\theta \mapsto J'(\Omega)(\theta)$ (from the Banach space $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into \mathbb{R}) via the $L^2(\partial\Omega)^d$ inner product.

Unfortunately, this choice is generally ill-suited, for at least two reasons:

- The deformation θ in (5.4) may be irregular, for instance because the calculations of u_Ω or p_Ω are polluted by numerical artifacts. The shape $\Omega_{\tau\theta}$ resulting from this deformation may thus show undesirable oscillations; see [138] for an illustration of this effect. It is often wise to proceed with a smoother deformation field than (5.4).
- The definition (5.4) only makes sense on the boundary $\partial\Omega$. Actually, the field v_Ω (and n) often has a natural extension to \mathbb{R}^d (for instance, the integrand in the shape derivative (4.30) makes sense everywhere inside Ω , since u_Ω and p_Ω do), but this natural extension has no legitimacy as a descent direction. We could probably find a “better” extension, depending on the targeted purpose.

Remark 5.1. *Incidentally, the choice (5.4) of a descent direction for $J(\Omega)$ is ill-suited from the mathematical viewpoint, since the space $L^2(\partial\Omega)^d$ is not a subspace of $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, and it is therefore not clear whether the variation Ω_θ of Ω can be defined in the framework of Hadamard’s method.*

Based on the considerations of Section 5.2.1, the idea of velocity extension and regularization of the $L^2(\partial\Omega)^d$ gradient of $J(\Omega)$ allows to overcome both issues. Let $H \subset W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ be a Hilbert space, with inner product $a(\cdot, \cdot)$. We solve the following identification problem

$$\text{Search for } g_\Omega \in H \text{ s.t. } \forall w \in H, \quad a(g_\Omega, w) = J'(\Omega)(w).$$

Obviously, $-g_\Omega$ is again a descent direction for $J(\Omega)$ if $g_\Omega \neq 0$, since for $\tau > 0$ small enough, the definition of shape derivative yields:

$$\begin{aligned} J(\Omega_{-\tau g_\Omega}) &= J(\Omega) - \tau J'(\Omega)(g_\Omega) + o(\tau) \\ &= J(\Omega) - \tau a(g_\Omega, g_\Omega) + o(\tau) \\ &< J(\Omega). \end{aligned}$$

Moreover, g_Ω is a “better” descent direction than $-v_\Omega n$, insofar as:

- The Hilbert space H is made of “regular” vector fields (they are at least in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$); therefore, the gradient $g_\Omega \in H$ is naturally “regular”.
- Since H contains vector fields defined on the whole ambient space \mathbb{R}^d (or, in practice on the whole computational domain D), then g_Ω is naturally defined on \mathbb{R}^d (resp. on D).

This paradigm paves the way to multiple interesting means to extract descent directions from the knowledge of the shape derivative $J'(\Omega)(\theta)$.

- The perhaps most natural choice is $H = H^m(\mathbb{R}^d)^d$, with the standard inner product:

$$(5.5) \quad a(u, v) = \sum_{|\alpha| \leq m} \int_{\mathbb{R}^d} \frac{\partial^\alpha u}{\partial x^\alpha} \cdot \frac{\partial^\alpha v}{\partial x^\alpha} dx.$$

The Sobolev embeddings indeed guarantee that, if $m > \frac{d}{2} + 1$, H is continuously embedded in $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ (see e.g. [2]). Functions in H are “regular” insofar as their derivatives up to order m are square integrable functions.

- In practice, one often takes $H = H^1(\mathbb{R}^d)^d$ which is not a subspace of $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$. However, as explained at the end of Section 5.2.1, if a little more structure is required from the shape derivative $J'(\Omega)(\theta)$, the latter can be extended to $\theta \in H$. This is the case if it reads as (5.3), with $v_\Omega \in L^2(\partial\Omega)$, which in turn holds true if u_Ω and p_Ω are in $H^2(\Omega)$; see Remark 4.4. In practice, this choice is very efficient, and very simple to program, although it is not completely rigorous from the mathematical viewpoint (see again Section 5.2.1).

When $H = H^1(\mathbb{R}^d)^d$ is chosen, one could also consider the following inner product in place of (5.5):

$$(5.6) \quad a(u, v) = \alpha^2 \int_{\mathbb{R}^d} \nabla u : \nabla v dx + \int_{\mathbb{R}^d} u \cdot v dx,$$

where α is to be interpreted as a regularization length-scale, indicating grossly speaking how far from the boundary $\partial\Omega$ (which is the support of $J'(\Omega)$) the regularized gradient is smeared.

- If in the considered optimization problem (2.9), a region Σ in space is required to belong to the boundary $\partial\Omega$ of shapes $\Omega \in \mathcal{U}_{\text{ad}}$, one could take:

$$H = \{v \in H^1(\mathbb{R}^d)^d, v = 0 \text{ on } \Sigma\},$$

with the inner product (5.5).

- Yet another (mathematically formal) possibility is to regularize the shape gradient (5.4) without extension outside from $\partial\Omega$. To this end, one may take $H = H^1(\partial\Omega)^d$, with the inner product inherited from the Laplace-Beltrami operator on H :

$$a(u, v) = \alpha^2 \int_{\partial\Omega} \nabla_T u : \nabla_T v ds + \int_{\partial\Omega} u \cdot v ds.$$

- In some applications, it is desirable to extend and regularize only the normal component v_Ω of the $L^2(\partial\Omega)^d$ gradient of $J(\Omega)$ in (5.4); more precisely, one searches for a descent direction under the form $\theta = -g_\Omega n$, where g_Ω is more regular than v_Ω and is defined on the whole space \mathbb{R}^d . To achieve this, one could consider the scalar Sobolev space $H = H^1(\mathbb{R}^d)$, with the inner product $a(\cdot, \cdot)$ corresponding to the scalar counterpart of (5.5). Then, one solves the variational problem:

$$\forall w \in H^1(\mathbb{R}^d), a(g_\Omega, w) = \int_{\partial\Omega} v_\Omega w ds.$$

- Finally, other choices are possible about the Hilbert space H of “regularized” deformations, such as $H = H^1(\mathbb{R}^d)^d$ equipped with the inner product inherited from the variational formulation of the linearized elasticity; see Section 6.2.2 below for an application example.

Let us finally pinpoint one last interest of this procedure. We have seen that the surface shape derivative $J'(\Omega)(\theta)$ in (5.3) has an alternative, volume form; see (4.23) and the discussion in the end of Section 4.5.3. We have argued that, although this last expression is explicit with respect to θ , it does not easily lend itself to the calculation of a descent direction. This is actually not exactly true. Indeed, using the present Hilbertian framework with for instance $H = H^1(\mathbb{R}^d)^d$, equipped with the inner product $a(\cdot, \cdot)$ in (5.6), and solving

$$\forall v \in H, a(g_\Omega, v) = J'(\Omega)(v)$$

with the volume expression of the shape derivative in the right-hand side makes total sense. It can be easily implemented in practice, and the resulting vector field $-g_\Omega$ is obviously a descent direction for $J(\Omega)$. This use of the distributed form of the shape derivative is very important in practice, and sometimes beats the use of the surface form (when the latter contains for instance high-order derivatives, or curvatures of the surface, etc, which are difficult to evaluate numerically). We refer to [106, 119, 189] about this idea.

This illustrates the important and general idea that the Hilbertian framework makes it possible to calculate descent directions from derivatives which have a very complicated form, as soon as they can be implemented as the right-hand side of a variational problem.

Remark 5.2. *The velocity regularization-extension of the shape gradient can be regarded as the mathematically rigorous counterpart of sensitivity filtering, an operation which is quite common in the context of density-based topology optimization [45].*

5.3. Constrained shape optimization

The algorithmic outline provided in Section 5.1 is devoted to unconstrained problems of the form (2.9), and it is natural to wonder how realistic situations, involving constraints, can be addressed. To set ideas, let us consider the shape optimization problem:

$$(5.7) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } C(\Omega) = 0,$$

featuring only one equality constraint. The perhaps simplest way to cope with the presence of a constraint in (5.7) is to introduce a Lagrangian function $\mathcal{L} : \mathcal{U}_{\text{ad}} \times \mathbb{R} \rightarrow \mathbb{R}$,

$$\mathcal{L}(\Omega, \ell) = J(\Omega) + \ell C(\Omega)$$

and to replace the minimization problem (5.7) by the following min-max or saddle point problem:

$$(5.8) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} \max_{\ell \in \mathbb{R}} \mathcal{L}(\Omega, \ell).$$

Indeed, problem (5.8) is equivalent to (5.7) because

$$\max_{\ell \in \mathbb{R}} \mathcal{L}(\Omega, \ell) = \begin{cases} J(\Omega) & \text{if } C(\Omega) = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Eventually, (5.8) can be solved by various methods, such as the *Uzawa* or the *Arrow-Hurwicz algorithms*. In brief, these optimization strategies proceed by intertwining, at each iteration $n \geq 0$ of the resolution (see e.g. [5, 25]),

- the minimization of an unconstrained and weighted average of $J(\Omega)$ and $C(\Omega)$:

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} \mathcal{L}(\Omega, \ell^n)$$

to find the next shape Ω^{n+1} ,

- the update of the Lagrange multiplier by a rule of the type $\ell^{n+1} = \ell^n + \delta^n C(\Omega^n)$ for some positive step $\delta^n > 0$.

This simple approach is reserved to academic situations, featuring only one (or a few) constraint because the convergence of the Lagrange multiplier ℓ , or equivalently the satisfaction of the constraints, is very slow and oscillatory.

We describe in this section how constrained optimization of the form (5.7), (or more complicated ones, featuring multiple equality or inequality constraints) can be addressed more efficiently in our framework. The cornerstone of these methods is the Hilbertian extension-regularization procedure described in Section 5.2. We start with the description of a simple and quite popular augmented Lagrangian algorithm, before turning to the implementation of an Sequential Linear Programming algorithm, quite in the spirit of [88]; see also [94]. Let us mention that other constrained optimization algorithms exist; see [38, 97].

5.3.1. A simple augmented Lagrangian algorithm

The augmented Lagrangian algorithm is a simple strategy to enforce one (or a few) equality or inequality constraint; it is easy to implement and it works “reasonably well” in practice. For simplicity, our presentation considers the problem (5.7) featuring only one equality constraint; we refer to e.g. [143] for a more general presentation.

The idea is again to convert (5.7) into a sequence of unconstrained problems (indexed by the superscript n) featuring two parameters $\ell \in \mathbb{R}$, $b \geq 0$:

$$(5.9) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} \mathcal{L}(\Omega, \ell^n, b^n), \text{ where } \mathcal{L}(\Omega, \ell, b) := J(\Omega) + \ell C(\Omega) + \frac{b}{2} C(\Omega)^2.$$

The augmented Lagrangian algorithm proceeds by alternating the resolution of (5.9) for fixed values of the parameters ℓ^n and b^n , and the update of the latter according to the rule, for some $\alpha > 1$:

$$(5.10) \quad \ell^{n+1} = \ell^n + b^n C(\Omega^n), \text{ and } b^{n+1} = \begin{cases} \alpha b^n & \text{if } b^n < b_{\text{target}}, \\ b^n & \text{otherwise.} \end{cases}$$

An informal justification of this process is as follows: b^n is a weight for a quadratic penalization of the constraint $C(\Omega)$; it is increased in the course of the optimization process, so that the constraint is enforced more and more strictly, but it does not need to tend to infinity (so that each intermediate optimization problem (5.9) is not dramatically ill-conditioned). The parameter ℓ^n is an increasingly accurate estimate of the Lagrange multiplier for the constraint $C(\Omega)$ in the first-order optimality conditions of (5.7).

In our applications, the computational burden of minimizing $\Omega \mapsto \mathcal{L}(\Omega, \ell, b)$ is significant, which suggests a more pragmatic version of the above idea: steps of the shape gradient Algorithm 5.1 for (5.9) are intertwined with updates of the coefficients ℓ^n, b^n ; see Algorithm 2.

Algorithm 2 Augmented Lagrangian algorithm for the constrained problem (5.7).

Initialization: Initial shape Ω^0 ; initial values of the parameters ℓ^0, b^0 .

for $n = 0, \dots$, until convergence **do**

- (1) Find a descent direction θ^n for $\Omega \mapsto \mathcal{L}(\Omega, \ell^n, b^n)$ as in Algorithm 1.
- (2) Obtain the new shape $\Omega^{n+1} := (\text{Id} + \tau^n \theta^n)$, where $\tau^n > 0$ is chosen sufficiently small so that $\mathcal{L}(\Omega^{n+1}, \ell^n, b^n) < \mathcal{L}(\Omega^n, \ell^n, b^n)$.
- (3) Calculate the new parameters ℓ^{n+1}, b^{n+1} from (5.10).

end for

return Ω^n

Remark 5.3. In the practical implementation of Algorithm 2, the parameter b^n is only increased every, say, 4 or 5 iteration, while ℓ^n is modified according to (5.10) at every step of the process.

The Augmented Lagrangian approach, however attractive for its simplicity, suffers at least two drawbacks. On the one hand, it is difficult to tune correctly its parameters (the initial values ℓ^0, b^0 , the amplification coefficient α, \dots), which turn out to be very case-dependent. On the other hand, the algorithm produces a lot of oscillations of the objective and constraint functions before they eventually reach their final values.

Remark 5.4. The same approach can be used to enforce an inequality constraint, up to a very classical trick in optimization practice, consisting in the addition of a “slack variable” [143]: for instance, a problem of the form

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } C(\Omega) \leq 0$$

is replaced by the augmented problem

$$\min_{\substack{\Omega \in \mathcal{U}_{\text{ad}}, \\ z \in \mathbb{R}}} J(\Omega) \text{ s.t. } C(\Omega) + z^2 = 0,$$

which can be solved owing to the previous methodology.

5.3.2. A Sequential Linear Programming method

A much more efficient strategy to deal with constrained optimization problems is linearize the objective and constraint functions, then to deduce from the obtained linear program a descent direction for the constrained non-linear problem. This *Sequential Linear Programming* (SLP) method is classical in shape optimization; see e.g. [88] in the context of the level set method. For the ease of the presentation, we consider a shape optimization problem featuring only inequality constraints:

$$(5.11) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } C_i(\Omega) \leq 0, \quad i = 1, \dots, p.$$

The main idea is to solve a series of constrained problems obtained by successive linearization of (2.6) around the current point. i.e. we solve:

$$(5.12) \quad \begin{aligned} \min_{\theta \in \Theta_{\text{ad}}} \quad & J(\Omega^n) + J'(\Omega^n)(\theta) \\ \text{s.t.} \quad & \begin{cases} C_i(\Omega^n) + C'_i(\Omega^n)(\theta) \leq 0, \quad i = 1, \dots, p, \\ \|\theta\|_{\Theta_{\text{ad}}} \leq \tau^n \end{cases} \end{aligned}$$

where Θ_{ad} is the admissible subset of $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ corresponding to \mathcal{U}_{ad} (see Section 4.1). Note that in (5.12), a *trust region* constraint is added about θ , whose role is to guarantee that this linearized version is a close approximation to the nonlinear version (5.11) when only deformations with norm smaller than the parameter τ^n are considered.

Inspired from the Hilbertian method of Section 5.2, we introduce a Hilbert space H with inner product $a(\cdot, \cdot)$, and we identify the shape derivatives in (5.12) to elements θ_J and θ_i in H , that is:

$$\forall v \in H, \quad a(\theta_J, v) = J'(\Omega^n)(v),$$

and for $i = 1, \dots, p$:

$$\forall v \in H, \quad a(\theta_i, v) = C'_i(\Omega^n)(v).$$

We then rewrite the problem (5.12) as:

$$(5.13) \quad \begin{aligned} \min_{\theta \in H} \quad & J(\Omega^n) + a(\theta_J, \theta) \\ \text{s.t.} \quad & \begin{cases} C_i(\Omega^n) + a(\theta_i, \theta) \leq 0, \quad i = 1, \dots, p \\ a(\theta, \theta) \leq (\tau^n)^2 \end{cases} \end{aligned}$$

which is a constrained problem in the Hilbert space H . The components of θ which are orthogonal to θ_J and θ_i in H do not play any role in this linear programming problem. Therefore, the descent direction θ is sought under the form:

$$\theta = \lambda_J \theta_J + \sum_{i=1}^p \lambda_i \theta_i,$$

for some coefficients $\lambda_J, \lambda_1, \dots, \lambda_p \in \mathbb{R}$ to be found. Our problem (5.13) then rewrites (omitting the trust region constraint for a moment):

$$(5.14) \quad \begin{aligned} \min_{(\lambda_J, \lambda_1, \dots, \lambda_p) \in \mathbb{R}^{p+1}} \quad & J(\Omega^n) + c^n \cdot \lambda \\ \text{s.t.} \quad & b^n + A^n \lambda \leq 0. \end{aligned}$$

where $b^n \in \mathbb{R}^p$, $c^n \in \mathbb{R}^{p+1}$ and $A^n \in \mathbb{R}^{p \times (p+1)}$ are the vectors and matrix defined by:

$$b^n = \begin{pmatrix} C_1(\Omega^n) \\ \vdots \\ C_p(\Omega^n) \end{pmatrix}, \quad c^n = \begin{pmatrix} a(\theta_J, \theta_J) \\ a(\theta_J, \theta_1) \\ \vdots \\ a(\theta_J, \theta_p) \end{pmatrix}, \quad \text{and} \quad A^n = \begin{pmatrix} a(\theta_1, \theta_J) & a(\theta_1, \theta_1) & \dots & a(\theta_1, \theta_p) \\ a(\theta_2, \theta_J) & a(\theta_2, \theta_1) & \dots & a(\theta_2, \theta_p) \\ \vdots & \vdots & \ddots & \vdots \\ a(\theta_p, \theta_J) & a(\theta_p, \theta_1) & \dots & a(\theta_p, \theta_p) \end{pmatrix}.$$

Let us finally discuss how the trust region constraint in (5.13) is incorporated in (5.14). Enforcing a bound on θ is not straightforward, while it is very simple for the coefficients λ_i . One simple procedure consists in requiring that:

$$|\lambda_J| \leq \frac{\Delta x}{2\|\theta_J\|_H}, \quad \text{and} \quad |\lambda_i| \leq \frac{\omega_i \Delta x}{2\|\theta_i\|_H}, \quad \text{where} \quad \omega_i = \frac{|C_i(\Omega^n)|}{\sum_{j=1}^p |C_j(\Omega^n)|}$$

as a measure of the relative violation of the i^{th} constraint. This simple approach guarantees that $\|\theta\|_H$ is of the order of Δx , the typical mesh size.

The linear, constrained optimization program (5.14), augmented with the adapted trust region constraint that we have just discussed, can now be solved efficiently by a standard solver such as `ipopt` [180], or `cvxopt` [32].

As already said, other constrained optimization algorithms are based on a linearization process, like the method of feasible directions [94] or the method of null space gradient flow [38, 97].

6. “BODY-FITTED” GEOMETRIC OPTIMIZATION

In this section, we describe one possible geometric optimization setting for the resolution of the unconstrained problem (2.9). The optimized shape Ω is represented by a mesh \mathcal{T} , which is deformed from one stage of the process to the next one. Despite its limitations, this historical, and still very popular, point of view is convenient for illustrating the previous concepts.

In order to ease the presentation of this section, we consider the model, unconstrained optimization problem (2.9), where the minimized function $J(\Omega)$ in (2.10) involves the solution u_Ω to the linearized elasticity system (2.1) on Ω .

6.1. Computational mesh and mesh deformation

The geometric shape optimization methods discussed in this section rely on a *simplicial mesh* \mathcal{T} of the optimized shape Ω , that is, a collection of open *simplices* $\{T_k\}_{k=1,\dots,N}$ (i.e. triangles in 2d, tetrahedra in 3d) which cover Ω in the sense that:

$$\bar{\Omega} = \bigcup_{k=1}^N \bar{T}_k.$$

In addition, we require that:

- The T_k do not overlap: $T_k \cap T_l = \emptyset$ whenever $k \neq l$;
- The mesh \mathcal{T} is *conforming* i.e. for all $k, l = 1, \dots, N$, the intersection $\bar{T}_k \cap \bar{T}_l$ is either a vertex, an edge, or a face (in 3d) of \mathcal{T} ;

see Figure 3 (right) for an illustration. Another crucial issue about \mathcal{T} is related to its *quality*. It is indeed well-known that the accuracy of finite element calculations performed using the mesh \mathcal{T} strongly depends on how close its elements are from being regular (i.e. with all edges of equal length); see e.g. [71, 90]. The quality of a mesh can be appraised by a variety of indicators, as we shall discuss a little more specifically in Section 6.2.1 below. More generally, we refer the reader to the comprehensive book [101] about meshing.

The application of the generic shape gradient Algorithm 1 to the resolution of the unconstrained problem (2.9) in the present context is summarized in Algorithm 3; it proceeds as follows. Each shape Ω^n arising in the course of the process is accounted for by a mesh \mathcal{T}^n . Thus, the calculation of the solutions u_{Ω^n} and p_{Ω^n} to the state and adjoint equations (Step 1) can be performed by using a standard finite element method on \mathcal{T}^n . Likewise, a descent direction θ^n from Ω^n is easily calculated in Step 3, be it via the naive rule (5.4) or thanks to the Hilbertian extension-regularization process (see Section 6.2.2 below).

Algorithm 3 Geometric optimization algorithm for problem (2.9).

Initialization: Initial shape Ω^0 , initial mesh \mathcal{T}^0 .

for $n = 0, \dots$, until convergence **do**

- (1) Calculate the solution u_{Ω^n} (resp p_{Ω^n}) to the state (resp. adjoint) equation owing to the finite element method on \mathcal{T}^n .
- (2) Calculate the shape derivative $\theta \mapsto J'(\Omega^n)(\theta)$ by using the formulas in Section 4.
- (3) Infer a descent direction $\theta^n : \mathbb{R}^d \rightarrow \mathbb{R}^d$ for $J(\Omega)$ from Ω^n .
- (4) Deform the mesh \mathcal{T}^n of Ω^n along θ^n for a short descent step $\tau^n > 0$ to obtain a new mesh \mathcal{T}^{n+1} of $\Omega^{n+1} = (\text{Id} + \tau^n \theta^n)(\Omega^n)$.

end for

return Ω^n

The delicate task in this framework is the deformation of the mesh \mathcal{T}^n involved in the update $\Omega^n \mapsto \Omega^{n+1} := (\text{Id} + \tau^n \theta^n)(\Omega^n)$ (Step 4), which is very likely to result in a low-quality, or even invalid mesh \mathcal{T}^{n+1} ; see Figure 12 below.

A number of numerical recipes make it possible to alleviate this issue, up to a certain point, as we discuss in the next section.

6.2. Avoiding mesh degeneracy in geometric methods

6.2.1. Local remeshing and mesh adaptation

Let us focus on one particular shape update step in Algorithm 3, omitting the reference to the iteration number n . Let $\Omega \subset \mathbb{R}^d$ be a shape equipped with a simplicial mesh $\mathcal{T} = \{T_k\}_{k=1,\dots,N}$ with vertices $\{x_i\}_{i=1,\dots,P}$; a vector field $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is given, which is in practice θ defined at each x_i , and we wish to deform \mathcal{T} into a new mesh \mathcal{T}^* of $\Omega^* := (\text{Id} + \theta)(\Omega)$, where we omit the descent step τ featured in Algorithm 3 for simplicity.

The “naive” way to realize this operation is to obtain \mathcal{T}^* by relocating every vertex x_i of \mathcal{T} at the position x_i^* pointed by θ without altering the connectivity of the mesh, i.e.

$$(6.1) \quad x_i \mapsto x_i^* := x_i + \theta(x_i), \quad i = 1, \dots, P.$$

Unfortunately, doing so is likely to result in a very low-quality mesh, not to say an invalid one, containing overlapping elements (see again Figure 12).

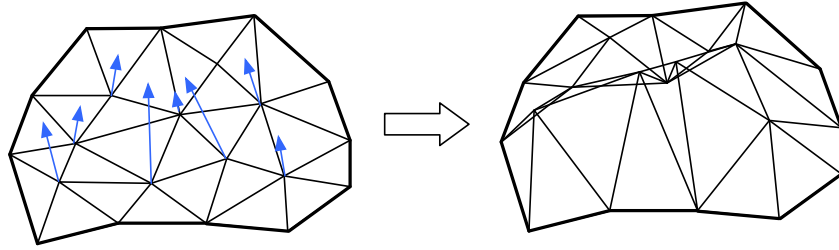


FIGURE 12. *Invalid deformation of a mesh.*

Following [83, 84], one simple strategy to cope with this difficulty is to proceed iteratively, intertwining stages where the motion (6.1) is realized “as far as possible”, and local *remeshing* stages, when a situation is encountered where elements are close to degenerate and the quality of the mesh has to be improved. This procedure is summarized in Algorithm 4 below; it unfolds as follows. At the beginning, the fraction $s = 0$ of the total motion (6.1) is accomplished; we search (e.g. by dichotomy) for the largest fraction $0 < \Delta s \leq 1$ of this total motion so that the operation

$$(6.2) \quad x_i \mapsto x_i^* := x_i + \Delta s \theta(x_i), \quad i = 1, \dots, P,$$

where the connectivities of the mesh are unaltered, yields a valid mesh $\tilde{\mathcal{T}}$. If $\Delta s = 1$, the procedure is complete and the new mesh is $\mathcal{T}^* = \tilde{\mathcal{T}}$. The contrary case where $\Delta s < 1$ indicates that $\tilde{\mathcal{T}}$ is ill-shaped: it contains nearly overlapping elements, whose *quality* is under a critical, security threshold. We then apply a local remeshing procedure to \mathcal{T}^* , relying on the following four operators which are illustrated in 2d on Figure 13.

- Edge split: a new vertex is introduced at the midpoint of a “too long” edge, and new simplices are created accordingly.
- Edge collapse: the endpoints of a “too short” edge in the mesh are merged, and the attached triangles are reconnected.
- Edge swap: this operation is significantly different in 2d and in 3d. In 2d, it consists in replacing the edge between two adjacent triangles with that connecting the two opposite vertices.
- Vertex relocation: one vertex is moved, while the connectivities of the mesh are unaltered.

This is done so that the quality of the resulting mesh \mathcal{T}^* has a user-defined minimum value. Starting from the new mesh \mathcal{T}^* , we can then go on with the motion (6.1) for the remaining fraction $(1 - \Delta s)$.

The efficiency of this procedure relies on a “good” measure of the quality of a mesh, in order to decide when to stop the motion to perform remeshing. In practice, the chosen quality factor must be severe enough to avoid the very “bad” elements, but also tolerant enough not to interrupt the process as soon as an elements

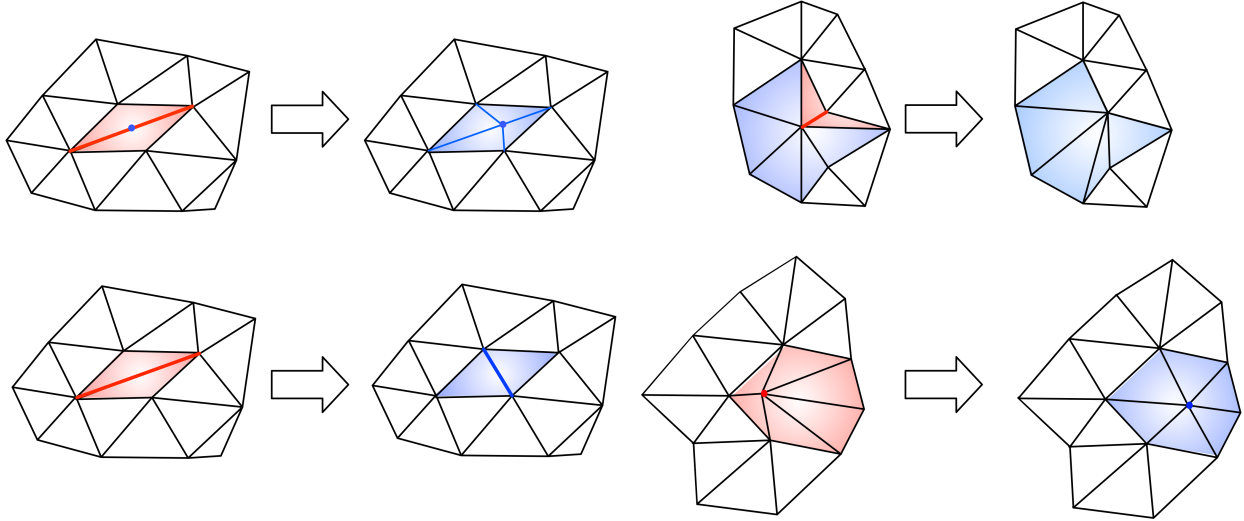


FIGURE 13. Local remeshing operations in 2d: (top, left) split of an edge; (top, right) collapse of an edge; (bottom, left) swap of an edge; (bottom, right) relocation of a vertex.

Algorithm 4 Mesh deformation algorithm.

Initialization: Initial mesh \mathcal{T} , $s = 0$.

while $s < 1$ **do**

- (1) Find by dichotomy the largest value Δs such that the mesh $\tilde{\mathcal{T}}$ resulting from (6.2) for a fraction Δs is valid, with quality above a given critical threshold.
- (2) Apply local remeshing operators to $\tilde{\mathcal{T}}$ so that the resulting mesh \mathcal{T}^* has good enough quality.
- (3) $\mathcal{T} = \mathcal{T}^*$;
 $s = s + \Delta s$.

end while

return \mathcal{T}^*

gets a little far from being regular (i.e. with all edges of equal length). A good trade-off for this purpose is offered by the quality function:

$$Q(T) = \alpha \frac{\text{Vol}(T)}{\left(\sum_{j=1}^{d(d+1)/2} |a_j|^2 \right)^{\frac{d}{2}}},$$

where the a_j are the $d(d+1)/2$ edges of the simplex T and the normalization factor α is taken so that $Q(T) = 1$ when T is regular. Roughly speaking, $Q(T)$ is close to 0 when T is nearly flat, and it is close to 1 when T is nearly equilateral.

We refer the interested reader to the open-source remeshing library `mng` [73] in which these features are implemented in two and three space dimensions; see [72] for the corresponding article.

Remark 6.1. *It is sometimes desirable to adapt the mesh to the solution u_Ω to the state equation for an improved resolution of the latter, or to use a finer mesh for the finite element resolution than that used to realize the motion (6.1), in order to ease the latter procedure.*

6.2.2. Taking advantage of a change of inner products

In the practical implementation of Algorithm 4, it is desirable that the descent direction θ inferred from the shape derivative $J'(\Omega)(\theta)$ “ease” the motion of the mesh \mathcal{T} of Ω according to the relocation rule (6.1), i.e. that it naturally tend not to cause its elements to degenerate.

Such an effect can be endowed to the descent direction thanks to the Hilbertian framework of Section 5.2. Let us for instance select $H = H_{\Gamma_D \cup \Gamma_N}^1(\Omega)^d$ as for the considered space of deformations, with the inner product inherited from linear elasticity:

$$\forall u, v \in H, \quad a(u, v) = \int_{\Omega} \left(Ae(u) : e(v) + u \cdot v \right) dx.$$

We recall from the definition (2.2) that descent directions $\theta \in H$ vanish on the fixed regions Γ_D and Γ_N during the optimization process (since they are not subject to optimization). Heuristically, descent directions calculated from this procedure should avoid compression of the simplices of \mathcal{T} , and so the degeneracy of the mesh, as a result of the physical properties of elastic displacements. This fairly well-established idea whereby elastic motions ease the process is presented for instance in [37, 142].

The above intuition can be reinforced by considering an inhomogeneous Hooke's tensor, taking constant values inside each element $T \in \mathcal{T}$:

$$\forall x \in T, \quad A(x) = \frac{1}{\eta_Q^2 + Q(T)} A,$$

where $Q(T)$ is the quality of T and $\eta_Q \ll 1$ is a very small parameter, avoiding division by 0. The rationale behind this formula is that the produced ‘‘elastic’’ descent direction θ corresponds to a situation where Ω is all the stiffer (i.e. it has a large Young's modulus) in mesh elements with low quality, so that they are less prone to get even more degenerate.

6.3. A numerical example: the benchmark cantilever test-case

To appraise the numerical behavior of the geometric shape optimization method presented in this section, we consider the benchmark *cantilever* test-case in two space dimensions, whose details are illustrated on Figure 14 (top, left). The optimized shape Ω is a beam, attached in a neighborhood Γ_D of its upper-left and lower-left corners; a vertical traction load $g = (0, -1)$ is applied on a region Γ_N at the middle of its right-hand side, and body forces f are omitted. We solve the unconstrained minimization problem:

$$(6.3) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) + \ell \text{Vol}(\Omega), \quad J(\Omega) = \int_{\Omega} Ae(u_{\Omega}) : e(u_{\Omega}) dx$$

of the compliance $J(\Omega)$ and a penalization of the volume $\text{Vol}(\Omega)$ of shapes by the fixed weight ℓ . The set \mathcal{U}_{ad} contains smooth shapes whose boundaries contain both regions Γ_D and Γ_N , and u_{Ω} is the elastic displacement given by (2.1).

The shape derivative of $J(\Omega)$ can be calculated along the lines of Section 4; it reads:

$$J'(\Omega)(\theta) = - \int_{\Gamma} Ae(u_{\Omega}) : e(u_{\Omega}) \theta \cdot n ds.$$

Note that the compliance functional is said to be *self-adjoint* since the adjoint state p_{Ω} introduced in this calculation is simply $p_{\Omega} = -u_{\Omega}$.

Some snapshots of the optimization process are reported on Figure 14. Interestingly, the boundary of the shape experiences quite large deformations, but its topology is not modified. In particular, it seems that the thin bar on the left-hand part of the shape would tend to break, but the body-fitted numerical representation of shapes does not allow such an operation. This observation is confirmed by the behavior of the convergence history, reported on Figure 15: obviously, the algorithm has not attained a local minimum at the final iteration.

6.4. Limitations and variants

In spite of the relative efficiency of the geometric optimization method presented in this section, this type of strategy should be reserved to situations where small changes are expected in the initial design. It particular, it does not allow for modifications of the topology of shapes (although we have just seen in the previous cantilever example of Figures 14 and 15 that it wants to do so). Nevertheless, the interest in geometric methods is far from extinct since there are many such problems, where the optimization process features small deformations of the shape and its topology is usually fixed, for example in aerodynamic shape design.

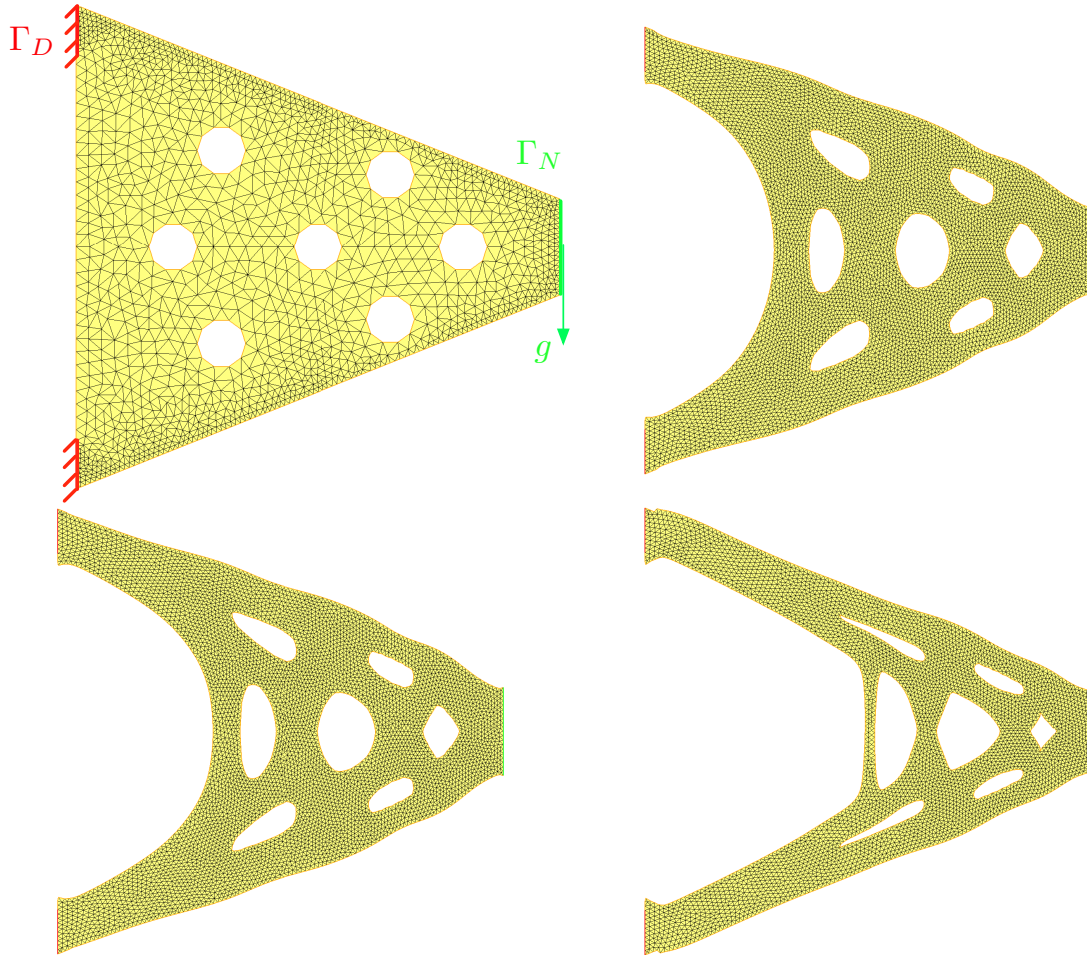


FIGURE 14. (From left to right, top to bottom) Iterations 0, 100, 120 and 200 of the cantilever geometric optimization test-case.

Moreover, an avatar of the previous algorithm, the so-called *Deformable Simplicial Complex* (DSC) method, was recently introduced in [70, 69], where an additional strategy based on the detection of very low-quality elements in the mesh, allows to detect and deal with ongoing topological changes.

Furthermore, the technical ingredients introduced in the present section (and notably the mesh modification operations described in Section 6.2.1) pave the way to the level-set based mesh evolution algorithm for shape and topology optimization introduced in [13], [14] and [96], which is described with more details in Section 8.

7. EULERIAN SHAPE CAPTURING TOPOLOGY OPTIMIZATION VIA THE LEVEL SET METHOD

In this section, we discuss the level set method for shape and topology optimization. In a nutshell, the optimized shape is represented as the subset where an auxiliary function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ takes negative values. In practice, ϕ is defined at the vertices of a *fixed* mesh \mathcal{T} of a large computational domain D , and the evolution of the shape is tracked via the evolution of these values. Thus circumventing the tedious remeshing issues plaguing “classical” geometric methods allows to account for dramatic changes of the optimized shape, including topological changes.

After presenting the general stakes of the level set method in Section 7.1, and related numerical algorithms in Section 7.2, we explain in Section 7.3 how it can be adapted to deal with shape optimization problems. Several numerical examples illustrating this strategy are presented in Section 7.4. In Section 7.5, we describe

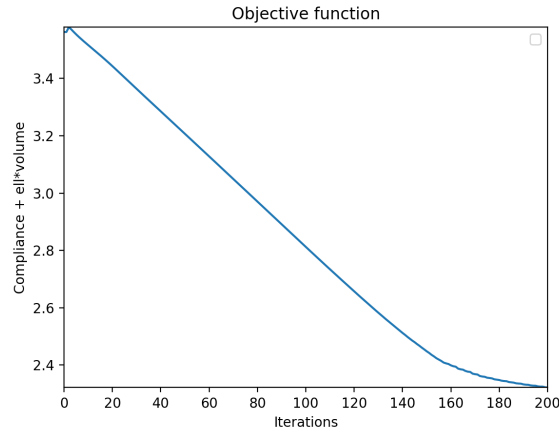


FIGURE 15. *Evolution of the minimized objective function in the cantilever geometric optimization test-case of Section 6.3.*

how topological derivatives can be incorporated into this workflow as a mechanism to nucleate holes inside the optimized shape. The final Section 7.6 contains a discussion about the main assets and drawbacks of this “classical” level set method.

7.1. The level set method for tracking the motion of a domain

The level set method was originally introduced by S. Osher and J. Sethian in the context of curvature flow [146]. It has then revealed an efficient tool in tracking the motion of an evolving boundary, e.g. in the context of fluid mechanics. We refer to the classical books [145, 162] for a comprehensive presentation (see also the chapter [159] in the same book).

The main philosophy behind the level set method is to trade the usual representation of a shape $\Omega \subset \mathbb{R}^d$ by the datum of a level set function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ which describes Ω as its negative subdomain; see Figure 16.

$$(7.1) \quad \forall x \in \mathbb{R}^d, \quad \begin{cases} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \partial\Omega, \\ \phi(x) > 0 & \text{if } x \in \mathbb{R}^d \setminus \bar{\Omega}. \end{cases}$$

This change in perspectives makes it quite simple to track the motion of a domain $\Omega(t)$, evolving in time

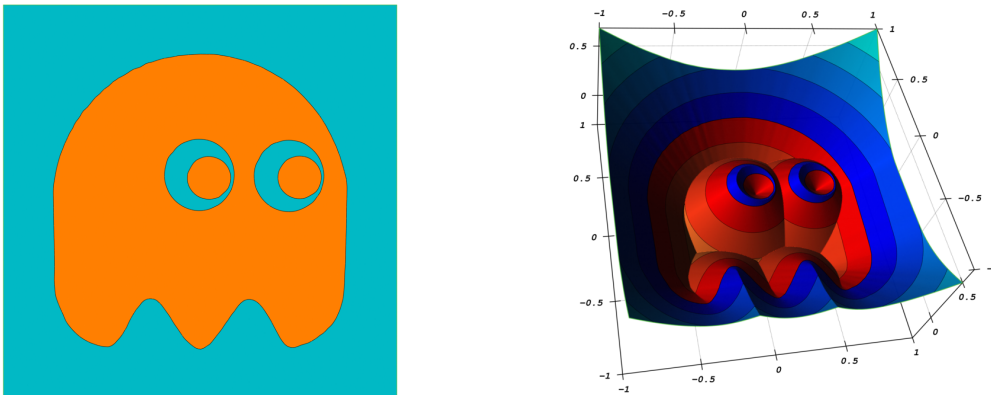


FIGURE 16. *(Left) A shape $\Omega \subset \mathbb{R}^2$ (orange); (right) graph of a level set function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ for Ω .*

according to a velocity field $V(t, x)$. Let indeed $\phi(t, \cdot)$ be a level set function for $\Omega(t)$, that is, (7.1) holds

at all times $t \in (0, T)$, where $T > 0$ is some prescribed final time; the evolution of $\Omega(t)$ translates into the following advection-like equation in terms of the function $\phi(t, x)$:

$$(7.2) \quad \frac{\partial \phi}{\partial t}(t, x) + V(t, x) \cdot \nabla \phi(t, x) = 0, \text{ for } (t, x) \in (0, T) \times \mathbb{R}^d.$$

Note that (7.2) is *not* a “true” advection equation since, in principle, the velocity field $V(t, x)$ depends on the moving domain $\Omega(t)$ itself, and so on $\phi(t, x)$. When this velocity happens to be oriented along the normal vector to $\Omega(t)$, i.e.

$$V(t, x) = v(t, x) \frac{\nabla \phi(t, x)}{|\nabla \phi(t, x)|},$$

for some scalar field $v(t, x)$, (7.2) becomes a Hamilton-Jacobi like equation:

$$(7.3) \quad \frac{\partial \phi}{\partial t}(t, x) + v(t, x) |\nabla \phi(t, x)| = 0, \text{ for } (t, x) \in (0, T) \times \mathbb{R}^d.$$

In the context of shape and topology optimization, the time variable in (7.2) and (7.3) is not the “true” time but rather some pseudo-time corresponding to the descent parameter in the minimization of the objective function. Likewise, the time step in the discretization of (7.2) and (7.3) is truly a descent step.

Remark 7.1. *One level set function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ as in (7.1) delimits two phases, namely the shape Ω defined by its negative subdomain, and the complement $\mathbb{R}^d \setminus \bar{\Omega}$, defined by its positive subdomain. Actually, using $n \geq 1$ level set functions ϕ_1, \dots, ϕ_n , it is possible to account for up to 2^n phases, namely those indexed by the 2^n subset of indices $I \subset \{1, \dots, n\}$ as follows:*

$$\left\{ x \in \mathbb{R}^d, \forall i = 1, \dots, n, \begin{cases} \phi_i(x) < 0 & \text{if } x \in I, \\ \phi_i(x) > 0 & \text{if } x \notin I \end{cases} \right\}.$$

The algorithms that we present below in the context of a single level set function are readily adapted to this context; see [182] for further details about this so-called color level set model.

7.2. Numerical algorithms for the level set method

In this section, we provide a brief overview of the numerical treatment of the most important operations involved in the implementation of the level set method.

Throughout this discussion, D is a bounded computational domain containing all the considered shapes Ω . It is equipped with a fixed mesh \mathcal{T} , about which we shall distinguish the following two situations:

- D is a box, and it is equipped with a Cartesian mesh: for example in 2d, its elements are rectangles with size $\Delta x \times \Delta y$ and vertices:

$$x_{ij} = (i\Delta x, j\Delta y), \quad i = 1, \dots, N_x, \quad j = 1, \dots, N_y.$$

- D is arbitrarily shaped, and \mathcal{T} is a simplicial mesh (see Section 6.1).

In both cases, the level set function ϕ is defined at the vertices of \mathcal{T} ; when its values are needed inside elements of \mathcal{T} , \mathbb{Q}_1 or \mathbb{P}_1 interpolation is used depending on the context.

7.2.1. Solving the equations of motion

In shape optimization, considering for instance the unconstrained minimization problem (2.9) of a function $J(\Omega)$, Algorithm 3 build a sequence of shapes Ω^n where each one is obtained by deformation of the previous one according to a shape gradient θ^n . In the limit of infinitely small descent steps (or time steps), this sequence Ω^n becomes a continuous time-dependent family of shapes $\Omega(t)$ and the sequence of shape gradients $\theta^n(x)$ becomes a velocity field $V(t, x)$ (recall that time is here a pseudo-time, corresponding to the descent parameter). Conversely, the practical resolution of (7.2) and (7.3) demands that the considered time period $(0, T)$ be divided into subintervals (t^n, t^{n+1}) , with a small enough time step $(t^{n+1} - t^n)$, and the sequence Ω^n is just a discrete approximation of $\Omega(t^n)$.

In the general context of the level set method, when dealing with realistic applications, the velocity field $V(t, x)$ depends in a highly non trivial way on the domain $\Omega(t)$, and so on the level set function $\phi(t, x)$ itself. Upon time discretization, with small enough intervals (t^n, t^{n+1}) , the practical resolution of (7.2) and

(7.3) relies on an explicit discretization of the velocity field $V(t, x)$ or $v(t, x)$, which is frozen over (t^n, t^{n+1}) , namely

$$\forall t \in (t^n, t^{n+1}), \quad V(t, x) \approx V(t^n, x), \quad \text{or} \quad v(t, x) \approx v(t^n, x).$$

Hence, the time discretization of (7.2) or (7.3) leads to a series of linear advection equations of the form

$$(7.4) \quad \begin{cases} \frac{\partial \phi}{\partial t}(t, x) + V(x) \cdot \nabla \phi(t, x) = 0 & \text{on } (0, T) \times D, \\ \phi(0, x) = \phi_0(x) & \text{on } D, \end{cases}$$

or “true” Hamilton-Jacobi equations:

$$(7.5) \quad \begin{cases} \frac{\partial \phi}{\partial t}(t, x) + v(x)|\nabla \phi(t, x)| = 0 & \text{on } (0, T) \times D, \\ \phi(0, x) = \phi_0(x) & \text{on } D, \end{cases}$$

over a generic period of time $(0, T)$, accounting for any of the above intervals (t^n, t^{n+1}) , and where $\phi_0(x)$ is a given initial condition; we now focus on this issue.

Remark 7.2.

- *Contrary to (7.4), (7.5) is nonlinear, and it is therefore more difficult to solve. Without entering into the details of the difficult underlying theory, let us solely mention that a proper notion of solutions has to be devised, that of viscosity solutions, which are continuous, but possibly non differentiable functions; see [108].*
- *The Hamilton-Jacobi equation (7.5) retains more information from the original equation (7.2), and notably the fact that the velocity field is consistently oriented in the normal direction. This is why the discretized process for solving (7.2) involving the Hamilton-Jacobi equations (7.5) is often preferred to its counterpart involving the advection equations (7.4).*

When the mesh \mathcal{T} is a Cartesian grid, finite difference techniques are available. The resolution by such means of the advection equation (7.4) is a fairly classical matter in numerical analysis, and we refer to e.g. [129]. The resolution of the Hamilton-Jacobi equation (7.5) is a little more intricate, and a particular care has to be paid to design *upwind* schemes, converging to the “good” solution (to be understood in the sense of viscosity solutions). Without entering into details, let us mention that first- and second-order schemes are provided in [162], §6.4. Higher-order schemes are obtained by using so-called (Weighted) Essentially Non Oscillatory finite differences for the approximation of the gradient term, which, in a nutshell, detect the regions where ϕ has to be smooth and those where, on the contrary, it tends to develop kinks; see [147]. As far as numerical algorithms are concerned, we mention the level set toolbox [135] on Cartesian meshes.

When the mesh \mathcal{T} is simplicial, these finite difference techniques are no longer available. There exist numerical methods to solve the Hamilton-Jacobi equation (7.5) in this context, but they are much more involved; let us mention in this direction Abgrall’s scheme [1] and the Petrov-Galerkin method of [39].

We now elaborate a little more on a general idea for solving the advection equation (7.4), which is relatively independent of the nature of the computational support: the *method of characteristics* [153] (see also Strain’s semi-Lagrangian method for (7.5) in [173]). Let us first recall that, under suitable regularity assumptions on the velocity field $V(x)$, the exact solution $\phi(t, x)$ to (7.4) is given by:

$$(7.6) \quad \phi(t, x) = \phi_0(X(0, t, x)), \quad t \in (0, T), \quad x \in D,$$

where $s \mapsto X(s, t, x)$ is the characteristic curve of V , emerging from x at time t , that is, the solution to the ordinary differential equation:

$$(7.7) \quad \begin{cases} \dot{X}(s, t, x) = V(X(s, t, x)) & \text{for } s \in (0, T), \\ X(t, t, x) = x. \end{cases}$$

Roughly speaking, $s \mapsto X(s, t, x)$ is the trajectory of a particle driven by the velocity field V , which is located at x at time t . Based on the formula (7.6), the calculation of $\phi(T, x)$ at an arbitrary vertex x of the computational mesh \mathcal{T} can be realized by solving the backward-in-time equation (7.7), e.g. thanks to a 4th-order Runge-Kutta scheme. We refer to [75] for an open-source implementation of this idea, and to [52] for the companion theoretical description.

Remark 7.3. When the motion of a domain $\Omega(t)$ is tracked by means of a level set function $\phi(t, x)$, only the values $\phi(t, \cdot)$ near the boundary $\partial\Omega(t)$ of the considered domain are needed, in principle. It seems therefore inefficient to store and update the values of $\phi(t, \cdot)$ everywhere inside the computational domain D at every time t . The so-called Narrow Band strategy precisely allows to track the motion of $\Omega(t)$ by only retaining a very small fraction of these values, namely those that are close to the 0 level set of $\phi(t, \cdot)$; see again [162] about this idea. Let us mention, nevertheless that when the level set method is used in a typical shape optimization workflow, the cost of these “inefficient” storage and update are in any event negligible with respect to other operations, such as the finite element calculations needed in the resolution of the physical equations.

7.2.2. Basic operations involving implicit domains

Let $\Omega \subset \mathbb{R}^d$ be a smooth domain, and let ϕ be a level set function for Ω , defined on the larger, hold-all or computational domain D . Most of the elementary operations involving Ω can be realized from the sole datum of ϕ , up to some adaptations that we briefly describe.

- The normal vector $n(x)$ to $\partial\Omega$, pointing outward Ω reads:

$$n(x) = \frac{\nabla\phi(x)}{|\nabla\phi(x)|}, \quad x \in D.$$

In a numerical implementation, one may approximate this vector as:

$$(7.8) \quad n(x) \approx \frac{\nabla\phi(x)}{(\eta_N^2 + |\nabla\phi(x)|^2)^{\frac{1}{2}}}, \quad x \in \partial\Omega.$$

where $\eta_N \ll 1$ is a very small positive parameter avoiding division by 0 (usually, we take $\eta_N \approx 10^{-10}$). Note that the above formula actually accounts for an extension of $n(x)$ from $\partial\Omega$ to the computational domain D as a whole.

- The second fundamental form $\Pi(x)$ and the mean curvature $\kappa(x)$ at the point $x \in \partial\Omega$ are the $d \times d$ matrix and real value defined by:

$$\forall \xi \in \mathbb{R}^d, \quad \Pi(x)\xi \cdot \xi = \nabla \left(\frac{\nabla\phi}{|\nabla\phi|} \right) (x) \xi \cdot \xi \quad \text{and} \quad \kappa(x) = \operatorname{div} \left(\frac{\nabla\phi}{|\nabla\phi|} \right) (x).$$

These quantities can be approximated as:

$$(7.9) \quad \Pi(x)\xi \cdot \xi \approx \nabla \left(\frac{\nabla\phi}{(\eta_N^2 + |\nabla\phi|^2)^{\frac{1}{2}}} \right) (x) \xi \cdot \xi \quad \text{and} \quad \kappa(x) \approx \operatorname{div} \left(\frac{\nabla\phi}{(\eta_N^2 + |\nabla\phi|^2)^{\frac{1}{2}}} \right) (x).$$

Again, since these formulas actually make sense for $x \in D$, they account for extensions of the second fundamental form and the mean curvature of $\partial\Omega$ from $\partial\Omega$ to D .

- Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a smooth function; we wish to calculate the integral

$$\int_{\Omega} f(x) \, dx = \int_D \chi_{\Omega}(x) f(x) \, dx,$$

where the characteristic function χ_{Ω} is defined by:

$$\forall x \in D, \quad \chi_{\Omega}(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

To do so, we fabricate an approximation of χ_{Ω} :

$$\forall x \in \mathbb{R}^d, \quad \chi_{\Omega}(x) \approx H_{\eta_H}(\phi(x)), \quad \text{where } H_{\eta_H}(t) := \frac{1}{2} \left(1 - \frac{t}{\sqrt{t^2 + \eta_H^2}} \right).$$

In the above, $H_{\eta_H}(t)$ is a smoothed version of the characteristic function of the interval $(-\infty, 0)$, in which the sharp transition between the values 1 and 0 is smeared over an interval of width $2\eta_H$; see Figure 17. A typical value for η_H is 10^{-3} . We then approximate:

$$\int_{\Omega} f(x) \, dx \approx \int_D H_{\eta_H}(\phi(x)) f(x) \, dx.$$

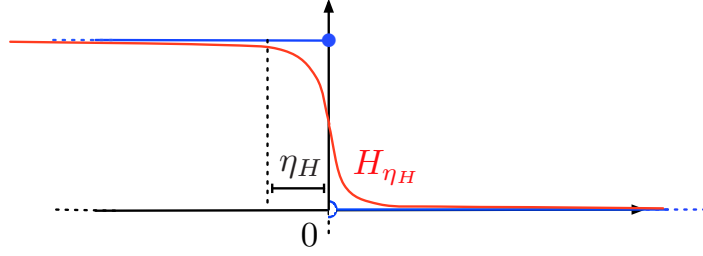


FIGURE 17. Smoothed characteristic function of the interval $(-\infty, 0)$ in \mathbb{R} .

- Let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ be a smooth enough function; we wish to calculate the boundary integral $\int_{\partial\Omega} g(x) ds(x)$. To this end, we rely on the fact that the distribution $\delta_{\partial\Omega} \in \mathcal{D}'(\mathbb{R}^d)$ accounting for the integration on $\partial\Omega$ reads, as a consequence of the Green's formula:

$$\delta_{\partial\Omega} = -\frac{\partial}{\partial n}(\chi_\Omega) \approx -\frac{\partial}{\partial n}(H_{\eta_H}(\phi)).$$

Hence, we obtain the approximate formula:

$$\int_{\partial\Omega} g(x) ds \approx \int_D \nabla(H_{\eta_H}(\phi(x))) \cdot n(x)g(x) dx.$$

- Finally, let Ω_1, Ω_2 be two domains in \mathbb{R}^d , described by the respective level set functions $\phi_1, \phi_2 : \mathbb{R}^d \rightarrow \mathbb{R}$. Then,

– one level set function ϕ_u for the reunion $\Omega_1 \cup \Omega_2$ reads:

$$\phi_u = \min(\phi_1, \phi_2),$$

– one level set function ϕ_i for the intersection $\Omega_1 \cap \Omega_2$ is:

$$(7.10) \quad \phi_i = \max(\phi_1, \phi_2),$$

– One level set function ϕ_c for the complement $\mathbb{R}^d \setminus \overline{\Omega_1}$ is:

$$\phi_c = -\phi_1.$$

7.2.3. Level set redistancing

In the theoretical context presented in Section 7.1, the level set method can proceed with an arbitrary level set function $\phi(t, x)$ for the evolving domain $\Omega(t)$. In practice, however, it is crucial that $\phi(t, x)$ be

- not too steep near the boundary $\partial\Omega(t)$, since this would entail inaccuracies in its localization; for instance, in the practice of the method of characteristics discussed in Section 7.2.1, a small error in the integration of (7.7) would then result in dramatic errors during the update of ϕ via the formula (7.6).
- not too flat near $\partial\Omega(t)$, since this would cause numerical instabilities in the evaluation of quantities such as the normal vector (7.8) or the mean curvature (7.9), due to division by near 0 values.

As was originally pointed out in [68], a good way to meet both requirements is to periodically restore the level set function into the *signed distance function* d_Ω :

$$\forall x \in \mathbb{R}^d, \quad d_\Omega(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega, \\ 0 & \text{if } x \in \partial\Omega, \\ d(x, \partial\Omega) & \text{if } x \in \mathbb{R}^d \setminus \overline{\Omega}, \end{cases}$$

where $d(x, \partial\Omega) := \min_{p \in \partial\Omega} |x - p|$ is the usual Euclidean distance from x to the boundary $\partial\Omega$. This can be achieved by means of a standard procedure for calculating the signed distance function to a domain such as the Fast Marching Method [161], or the Fast Sweeping Method [185]; see also [122] and [157] for corresponding strategies when the computational mesh of D is simplicial, and [74, 75] for an open-source implementation.

An alternative way to calculate the signed distance function d_Ω when a level set function ϕ is available for Ω (which is possibly very steep or very flat) is to solve the so-called redistancing equation to steady state:

$$(7.11) \quad \begin{cases} \frac{\partial \psi}{\partial t}(t, x) + \operatorname{sgn}(\phi)(x)(|\nabla \psi(t, x)| - 1) = 0 & \text{for } t > 0, \ x \in \mathbb{R}^d, \\ \psi(0, x) = \phi(x) & \text{in } \mathbb{R}^d; \end{cases}$$

see again [68], or [36] for the mathematical foundations of this idea.

Remark 7.4. *Having available the signed distance function d_Ω to the considered shape Ω , and not just any associated level set function, is desirable for many practical purposes beyond that of ensuring a better numerical stability of the level set method. For instance, we shall see in Section 9.2 that many delicate geometric features of Ω , such as its minimum and maximum thicknesses, are conveniently formulated with the help of d_Ω .*

7.3. Shape and topology optimization using the level set method

The level set method was first considered in the shape optimization context in [148] for a two-phase conductivity equation and in [163] for the elasticity system, but without the notion of shape derivative. The coupling of the level set machinery with the method of Hadamard was introduced in structural optimization in [21, 22, 183] in a systematic fashion. Considering again the constraint-free problem (2.9) in the physical context of linear elasticity of Section 2.1.1, we illustrate how the generic shape optimization workflow of Section 5.1 can benefit from a numerical representation of shapes by the level set method. This combination gives rise to Algorithm 5 below, which we now explain with a little more details.

All the considered shapes Ω lie in a computational domain D (e.g. a box), which is equipped with a fixed (Cartesian, or simplicial) mesh \mathcal{T} . Any shape $\Omega \subset D$ arising in the course of the evolution process is described by a level set function ϕ , which is supplied by its values at the vertices of \mathcal{T} . For simplicity, the non optimizable regions Γ_D and Γ_N of $\partial\Omega$ where respectively the shapes are clamped and loads are applied, are also subsets of ∂D . In truth, if the loaded boundary Γ_N is fixed and not subject to optimization, the clamped boundary Γ_D is not treated exactly in the same way, as we now explain. A subset ∂D_D of the boundary of the computational domain D is fixed once and for all. Then, the clamped boundary Γ_D of any shape Ω is free to be any subset of ∂D_D . In other words, Γ_D cannot move in the normal direction to ∂D but it can slide tangentially to ∂D , as long as it stays inside ∂D_D . If the solution u_Ω is smooth at the border of Γ_D , there is no further term to add to the shape derivative, compared to the case where Γ_D is fixed.

The uneasy task in this framework is the resolution of the state equation (2.1) (and similarly for the adjoint equation), since no mesh of the shape is available. Often, (2.1) is approximated by a partial differential equation posed on D as a whole. In the present situation where the optimized boundary Γ bears homogeneous Neumann boundary conditions, the so-called *ersatz material approximation* is used, whereby the void $D \setminus \Omega$ is filled with a very soft material, with Hooke's tensor εA , for a very small parameter $\varepsilon \ll 1$ (typically, $\varepsilon \approx 10^{-3}$). To be precise, the exact elastic displacement u_Ω , defined by (2.1), is replaced by the unique solution $u_{\Omega, \varepsilon} \in H_{\Gamma_D}^1(D)^d$ to:

$$(7.12) \quad \begin{cases} -\operatorname{div}(A_\varepsilon e(u_{\Omega, \varepsilon})) = 0 & \text{in } D, \\ u_{\Omega, \varepsilon} = 0 & \text{on } \Gamma_D, \\ A_\varepsilon(u_{\Omega, \varepsilon})n = g & \text{on } \Gamma_N, \\ A_\varepsilon(u_{\Omega, \varepsilon})n = 0 & \text{on } \partial D \setminus (\overline{\Gamma_D} \cup \overline{\Gamma_N}), \end{cases}$$

where

$$A_\varepsilon(x) = \chi_\Omega(x)A + (1 - \chi_\Omega(x))\varepsilon A = \begin{cases} A & \text{if } x \in \Omega, \\ \varepsilon A & \text{if } x \in D \setminus \Omega. \end{cases}$$

In practice, the characteristic function of Ω in the definition of A_ε is calculated thanks to the methods of Section 7.2.2. A similar approach is applied to the calculation of an approximate adjoint state $p_{\Omega, \varepsilon}$.

This ersatz material approximation is consistent: one can indeed prove that, for a given shape Ω , the following convergence holds:

$$\|u_{\Omega, \varepsilon} - u_\Omega\|_{H^1(\Omega)^d} \xrightarrow{\varepsilon \rightarrow 0} 0.$$

and a similar approach can be applied for the calculation of the adjoint state p_Ω .

The execution of the other steps of the shape optimization Algorithm 1 in the level set framework do not entail any particular difficulty. Notably, the thorny shape update operation (Step 4 in Algorithm 1) is made simple and robust thanks to the use of the level set method. Let us observe in passing that the use of the level set method for this shape update requires the shape gradient to be defined everywhere on D , which is not an issue if the velocity extension-regularization method of Section 5.2 is used. Note that, depending on whether (7.4) or (7.5) is solved when tracking the motion of the shape, this procedure involves a Hilbert space H composed of either vector fields V (or θ), or scalar-valued normal velocities v , respectively. All in all, this leads to Algorithm 5 where we deliberately chose not to discuss the choice of a convergence criterion, which is usually delicate since it has to account for a balance between a small total number of iterations and a precise convergence. However, let us say a few words about the choice of the descent or time step τ^n . If the numerical scheme for solving the Hamilton-Jacobi equation in Step 3 of Algorithm 5 entails a limitation on the size of its time step (like a CFL condition), one should not restrict τ^n by this limitation. Rather, several “internal” time steps should be performed for solving the Hamilton-Jacobi equation until time τ^n , which is chosen as a descent step. Namely, τ^n is monitored by the decrease of $J(\Omega^n)$. A typical simple strategy is to (moderately) increase τ^n each time the objective function decreases, while it is halved each time the objective function increases (and the current iteration is rejected).

Algorithm 5 Level set shape optimization method for problem (2.9).

Initialization: Computational domain D , equipped with the mesh \mathcal{T} , level set function ϕ^0 for the initial shape $\Omega^0 \subset D$.

for $n = 0, \dots$, until convergence **do**

- (1) Calculate the solution $u_{\Omega^n, \varepsilon}$ to the approximate state equation (7.12), and if needed that $p_{\Omega^n, \varepsilon}$ to the (approximate) adjoint equation.
- (2) Use the derivative $J'(\Omega^n)(\theta)$ (see Section 4) to infer a (scalar) descent direction for $J(\Omega)$ by solving the (scalar) extension-regularization problem:

$$\forall v \in H, \quad a(v_{\Omega^n}, v) = -J'(\Omega^n)(v n_{\Omega^n}),$$

where for instance:

$$H = H_{\Gamma_D \cup \Gamma_N}^1(D), \quad \text{and } a(u, v) = \alpha^2 \int_D \nabla u \cdot \nabla v \, dx + \int_D uv \, dx.$$

- (3) Solve the level set Hamilton-Jacobi equation:

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + v_{\Omega^n}(x) |\nabla \phi(t, x)| = 0 & \text{for } t \in (0, \tau^n), \quad x \in D, \\ \phi(0, x) = \phi^n(x) & \text{for } x \in D, \end{cases}$$

over a suitably short period of time $(0, \tau^n)$.

- (4) The new shape is defined by: $\Omega^{n+1} := \{x \in D, \phi(\tau^n, x) < 0\}$.

end for

return Ω^n

Remark 7.5. *The above ersatz material approximation is only one means, among many, to evaluate the solution u_Ω to the linear elasticity system (2.1) when Ω is not equipped with a mesh of its own, but is instead supplied by the values of a level set function at the vertices of the mesh of D . This simple approximation is reasonably accurate in practice, but it has undesirable effects in some cases, such as the low accuracy of stress evaluation near the boundary (see e.g. [19]) or the appearance of thin ligaments in compliant mechanism design. A simple way to avoid some of these problems is to remove the void elements from the discretized system; besides, doing so drastically cuts down the computational cost. Another approach consists in adapting the ersatz parameter during the computation, starting with moderately low values of ε and making it gradually tend to zero. It has been successfully used in the quite sensitive design of compliant mechanisms (see e.g. Figure 26 below). Note that, for eigenvalue optimization, one has to define an ersatz density which should be much smaller than the ersatz rigidity for consistency issues (see [18]). Finally, various strategies exist for a precise resolution of (2.1) in this fixed mesh context, at the expense of additional implementation difficulties;*

let us mention *X-FEM* [89, 179] or *cutfem* [56] methods, in which the finite element space is enriched with additional basis functions near the interface $\partial\Omega$.

7.4. Numerical examples

7.4.1. The benchmark two-dimensional cantilever test-case revisited

We first go over the 2d cantilever test-case of Section 6.3, with the help of the “classical” level set method presented in this section. The considered shapes are enclosed in a rectangle D with size 2×1 ; they are clamped on the left-hand part of their boundary, and a unit vertical point load $g = (0, -1)$ is applied in the middle of their right-hand side; see Figure 18 (top, left).

In this context, we solve the same problem (6.3) as in Section 6.3, that of the unconstrained minimization of a weighted sum of the compliance and the volume of shapes. Several stages of the optimization process are displayed on Figure 18, and the associated convergence history is that of Figure 19.

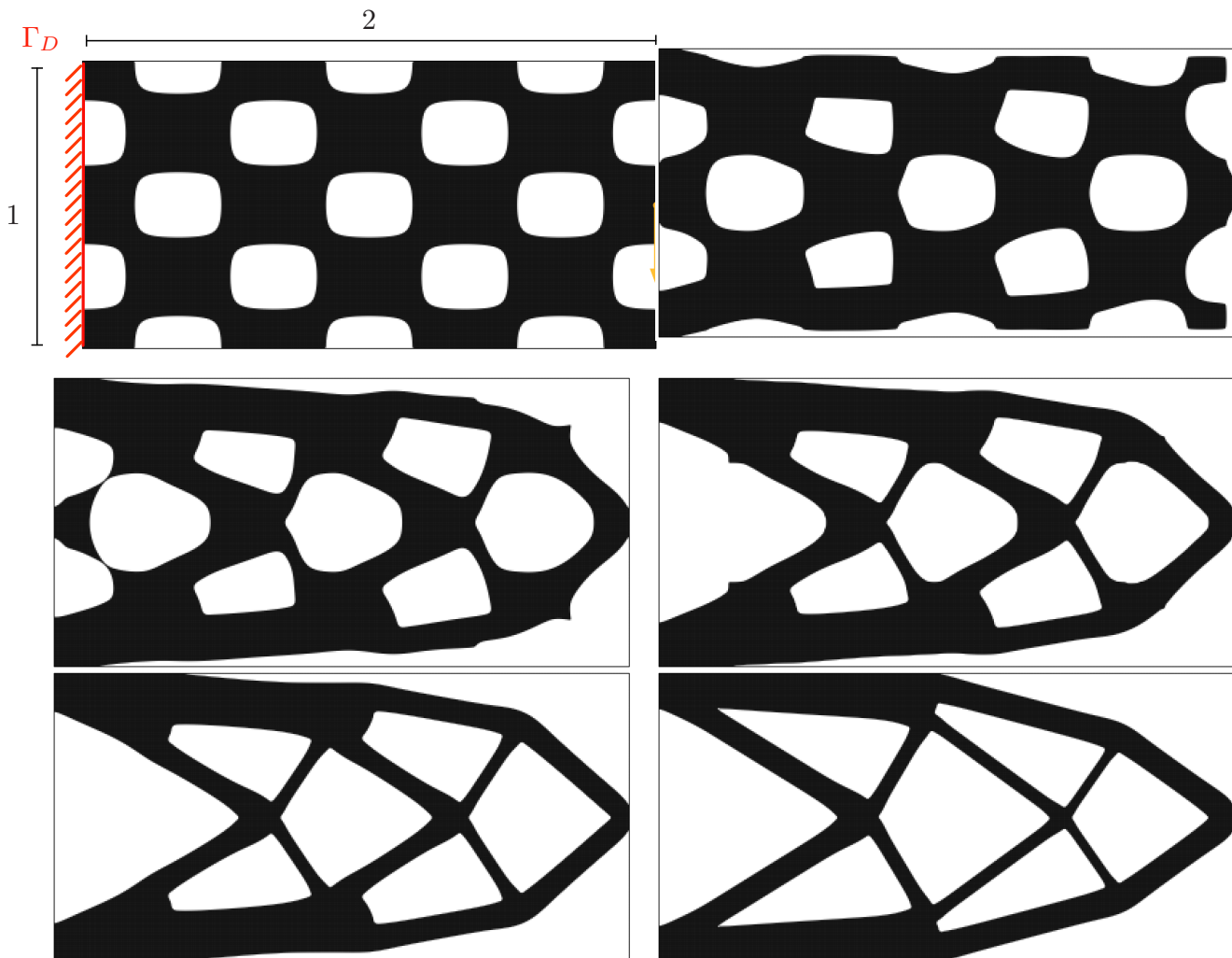


FIGURE 18. (From left to right, top to bottom) Optimization of the 2×1 cantilever with the level set method in Section 7.4.1: initial guess and details of the test-case, iterations 5, 10, 15, 25 and 200 (final shape).

We note that the shape changes topology on several occurrences in the course of the evolution. This is a priori surprising since we would expect from the use of the method of Hadamard to result in successive shapes which are diffeomorphic to each other (see notably Proposition 4.1). That such is not the case is actually

due to the fact that we are abusing the framework of Hadamard: each updated shape Ω^{n+1} is obtained from the previous one Ω^n via the relation $\Omega^{n+1} = (\text{Id} + \tau^n \theta^n)(\Omega^n)$, where θ^n is the descent direction from Ω^n , and τ^n is the chosen time step. In practice, we do not verify that the norm $\tau^n \|\theta^n\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)}$ of the actual deformation is lower than 1, which is the condition for Ω^n and Ω^{n+1} to be diffeomorphic. The level set method allows to realize the deformation even though topological changes are involved, and we control that the minimized objective function indeed decreased from Ω^n to Ω^{n+1} . In practice, we only impose that this decrease hold up to a certain tolerance. This allows the optimized shape to escape from local minima thanks to a change of topology. This phenomenon is clearly observed in the convergence history of Figure 19, where the objective function increases slightly on several occasions between iterations 10 and 90, even though it ends up globally decreased at the end of the process. Comparing this result with that obtained in Section 6.3 by applying a geometric shape optimization algorithm to the very same example is instructive (see Figure 14): in the latter case, the optimization process has ended with a “bad” local minimum because no mechanism allowed the bar on the left-hand side of the shape to break.

This example illustrates why the level set method is labelled as a method for shape and topology optimization, even though the underlying Hadamard definition of shape derivatives does not, in theory, leave the room for topological changes. Note finally that with this abuse of the method of Hadamard, we are only able to merge holes, but not to nucleate new ones within the bulk structure, at least in 2d; see Section 7.5 for a different mechanism achieving this purpose. In 3d holes and handles can easily appear by pinching thin material walls (see the numerical examples below in Figures 23, 25, 26).

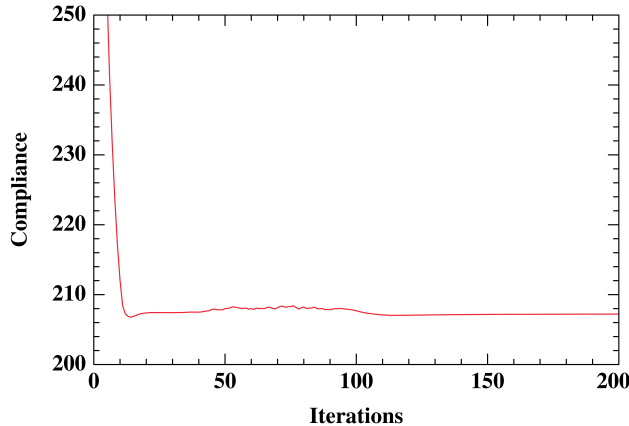


FIGURE 19. *Convergence history of the minimized function in the 2d cantilever example of Section 7.4.1.*

7.4.2. A two-phase optimization example

Minor adaptations to the framework presented in Section 7.3 allow to deal with the optimization of more than two phases, as we now briefly explain, referring to [11] for full details. We consider the setting in Figure 20: a two-dimensional box D with size 2×1.2 is attached at its lower-left and lower-right corners; a vertical force with value $g = (0, -1)$ is applied on two different regions of the bottom side, while a force $2g$ is applied at the center of the bottom side.

The domain D is filled with four elastic materials with different properties, identified by the four Hooke’s tensors A_i , $i = 1, 2, 3, 4$. Obviously, relying on the ersatz material approximation, one or several of these phases may account for void if the corresponding Hooke’s tensor A_i has very small Young’s modulus $E_i = \varepsilon \ll 1$. The four phases are parametrized by two subdomains $\Omega_1, \Omega_2 \subset D$, so that the Hooke’s tensor A_{Ω_1, Ω_2} accounting for the material properties inside D reads (see also Remark 7.1):

$$(7.13) \quad \forall x \in D, \quad A_{\Omega_1, \Omega_2}(x) = \begin{cases} A_1 & \text{if } x \in \Omega_1 \cap \Omega_2, \\ A_2 & \text{if } x \in (D \setminus \overline{\Omega_1}) \cap \Omega_2, \\ A_3 & \text{if } x \in \Omega_1 \cap (D \setminus \overline{\Omega_2}), \\ A_4 & \text{if } x \in (D \setminus \overline{\Omega_1}) \cap (D \setminus \overline{\Omega_2}). \end{cases}$$

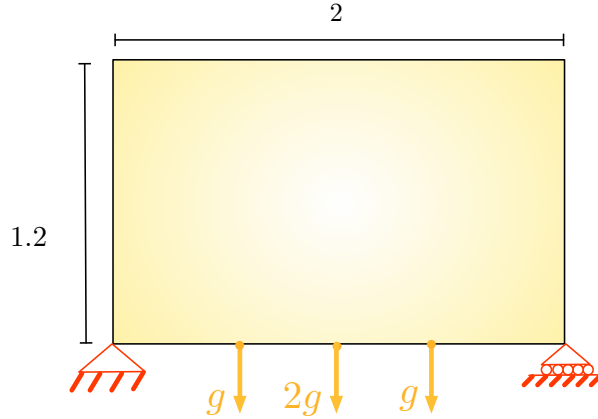


FIGURE 20. *Setting of the multi-material three-force bridge in Section 7.4.2.*

The elastic displacement u_{Ω_1, Ω_2} of the total structure D is then described by the counterpart of (2.1) posed on D , where the Hooke's tensor is A_{Ω_1, Ω_2} , as defined by (7.13). In this context, we minimize the compliance

$$J(\Omega_1, \Omega_2) = \int_D A_{\Omega_1, \Omega_2} e(u_{\Omega_1, \Omega_2}) : e(u_{\Omega_1, \Omega_2}) \, dx$$

as a function of the two shapes Ω_1, Ω_2 . A target value $V_{T,i}$ is imposed for the volume of the i^{th} phase by means of the Augmented Lagrangian strategy described in Section 5.3.1.

Two numerical examples are shown in Figure 21. In the first one, only three materials are considered (one being void): the Young's moduli of the phases A_i read $E_1 = 1$, $E_2 = 0.5$, and $E_3 = E_4 = \varepsilon$. The volume targets are $V_{T,1} = 0.2|D|$ and $V_{T,2} = 0.1|D|$. In the second example, we consider the repartition of four materials (one being void): $E_1 = 1$, $E_2 = 0.5$, $E_3 = 0.25$ and $E_4 = \varepsilon$, with the volume targets $V_{T,1} = V_{T,2} = V_{T,3} = 0.1|D|$, so that the same amount of non void material is used in both situations. Obviously, the resulting material repartitions are quite different, but in both cases, the strongest material is placed near the loading regions and on the outer outline of the structure.

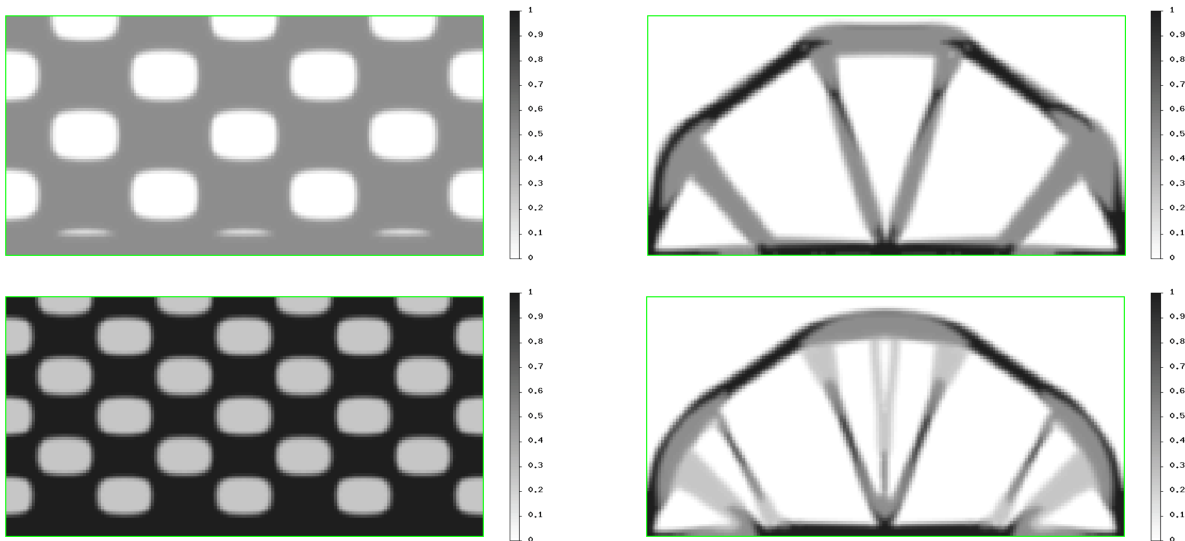


FIGURE 21. *(Upper row) Initial and optimized designs of a 3-force bridge using two materials and void; (lower row) initial and optimized designs using three materials and void.*

7.4.3. Various 3d examples on fixed meshes

In this section, we present several three-dimensional examples in the structural mechanics context of Section 2.1.1, which demonstrate the efficiency of the “Eulerian” level set method for shape and topology optimization introduced in this section.

Our first example is a 3d version of the cantilever case from Sections 6.3 and 7.4.1. The setting is depicted in Figure 22 (top, left): the computational domain is a $1 \times 2 \times 1$ box; shapes are clamped on the rear face $\{x = (x_1, x_2, x_3) \in \mathbb{R}^3, x_2 = 0\}$, and a unit vertical point load $g = (0, 0, -1)$ is applied in the middle of the opposite face $\{x = (x_1, x_2, x_3) \in \mathbb{R}^3, x_2 = 2\}$. In this context, we minimize the compliance (2.7) of the shape Ω with a constraint on its volume $\text{Vol}(\Omega)$ which must be equal to $|D|/4$. This volume constraint is imposed via the Augmented Lagrangian method of Section 5.3.1. Using the symmetry of the situation with respect to the plane $\{x = (x_1, x_2, x_3) \in \mathbb{R}^3, x_1 = 0\}$, only one half of the domain is considered during the optimization, which is meshed with $35 \times 140 \times 70 = 343000$ hexahedral elements. Starting from an initial shape with multiple holes, 200 iterations of the level set based shape optimization Algorithm 5 are performed, and several intermediate shapes are reported in Figure 22; interestingly, the intermediate shapes show various complex topologies, even though the final solution has a quite simple one.

Our second example is concerned with the optimization of a three-dimensional bridge. The shapes are contained inside a domain D shaped as a trapezoid with dimensions $4 \times 1 \times 1$ (see Figure 23, (top, left)); shapes are clamped on the two lateral boundaries of D , and a vertical load is applied on the upper part of ∂D , accounting for the effect of pedestrians or cars. Starting from the full domain D as initial shape, the compliance (2.7) of Ω is minimized under the constraint that its volume be $\text{Vol}(\Omega) = |D|/8$. Taking advantage of the symmetry of the problem, only $1/4$ of D is discretized, by a mesh consisting of $120 \times 30 \times 60 = 216000$ hexahedral elements. Several intermediate stages of the computation are reported on Figure 23. In particular, we observe an opposite behaviour in the shape optimization process when compared to that in the 3d cantilever test case: the initial guess has no holes while the final design shows a very complex topology, without any use of the topological derivative.

Note that the computational domain D in this case is not a box, which could, at first sight, prevent us from using a Cartesian mesh for its discretization. In practice, we rely on a given Cartesian mesh of the (larger) $4 \times 1 \times 1$ bounding box \tilde{D} (whose edges are in green in Figure 23), and we rely on an additional, given, level set function $\tilde{\phi}$ for the domain D to enforce shapes to be subset of D ; more precisely, at each stage of the optimization process, the level set function ϕ for the optimized shape is defined on the large domain \tilde{D} , and the intersection of the corresponding negative subdomain with D is simply calculated by using (7.10).

Pursuing this idea, we now exemplify how it is possible, even when Cartesian meshes are used, to handle complex geometries for the hold-all or computational domain D . In Figure 24 three different optimized cantilever beams are displayed (right-hand column), in the same setting as that considered at the beginning of this section, except that different hold-all domains D are used (left-hand column).

Another illustration of the capabilities of the algorithm to easily capture complex 3d topologies is given on Figure 25 with different optimal chairs obtained from a full domain initialization. Uniform perpendicular loads are applied on the seat and back of the chair, the vertical displacement vanishes on the bottom boundary and compliance is minimized with a volume constraint.

Finally, we turn to the optimal design of a *compliant mechanism*. These are monolithic (i.e. jointless) devices converting an imposed force or displacement on an “input” port into a prescribed displacement on a different, “output” port. The setting is that of a 3d gripping mechanism, as depicted on Figure 26 (top): the computational domain D is a box with size $1 \times 1 \times 1$. A displacement u_d is imposed on the “input” regions ω_{in} (in blue), and the optimized shape is expected to show maximum displacement in the given direction $d \in \mathbb{R}^3$ in the “output” regions ω_{out} (in red). Mathematically, the shape optimization problem is that of

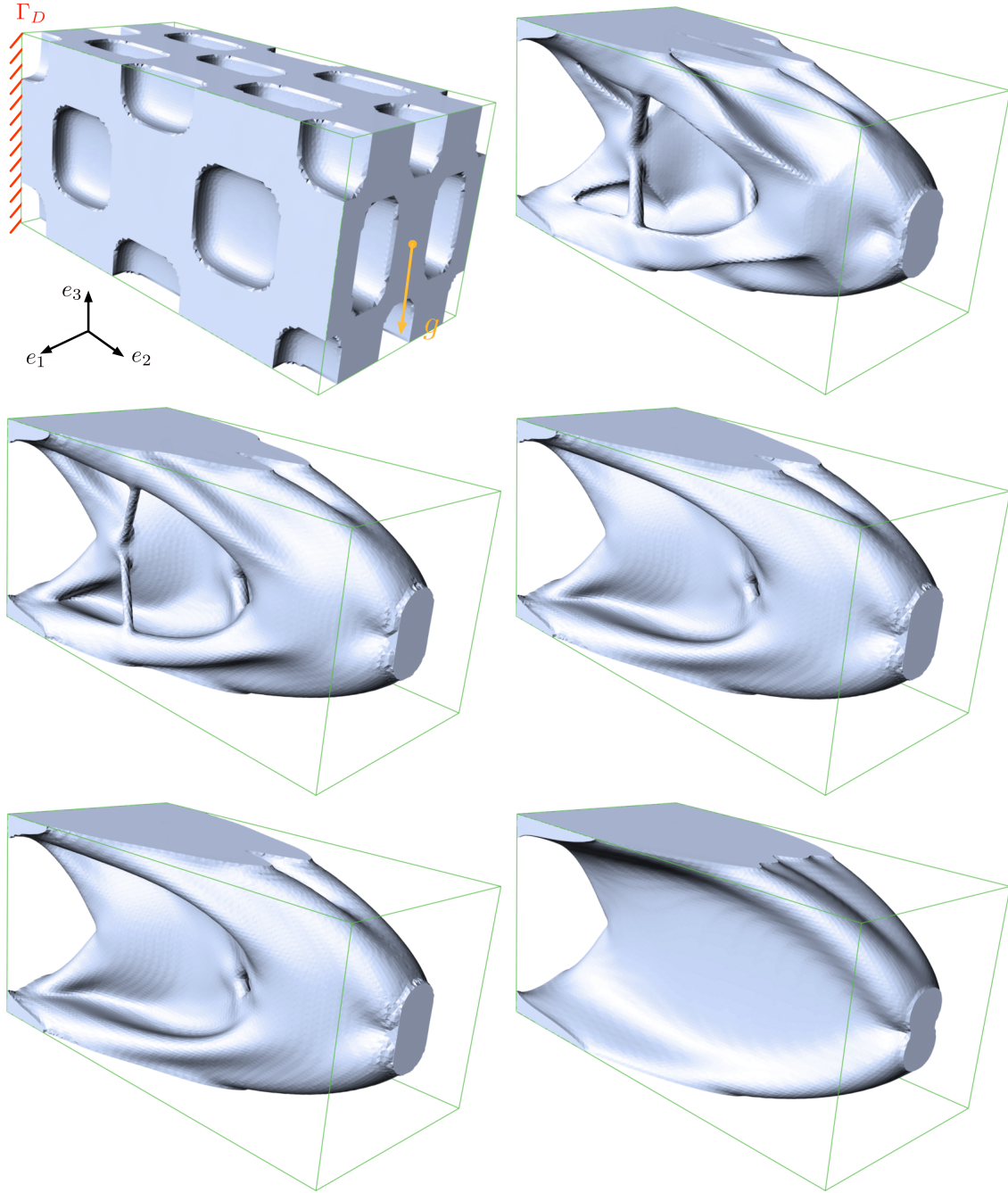


FIGURE 22. (From left to right, top to bottom) Optimization of a 3d cantilever: initial guess with details of the test-case, iterations 30, 40, 50, 60 and 200 (final shape).

minimizing, without constraint, the so-called “geometric advantage” criterion:

$$J(\Omega) = - \frac{\int_{\omega_{\text{out}}} |u_{\Omega}(x) \cdot d|^2 dx}{\int_{\omega_{\text{in}}} |u_d(x)|^2 dx};$$

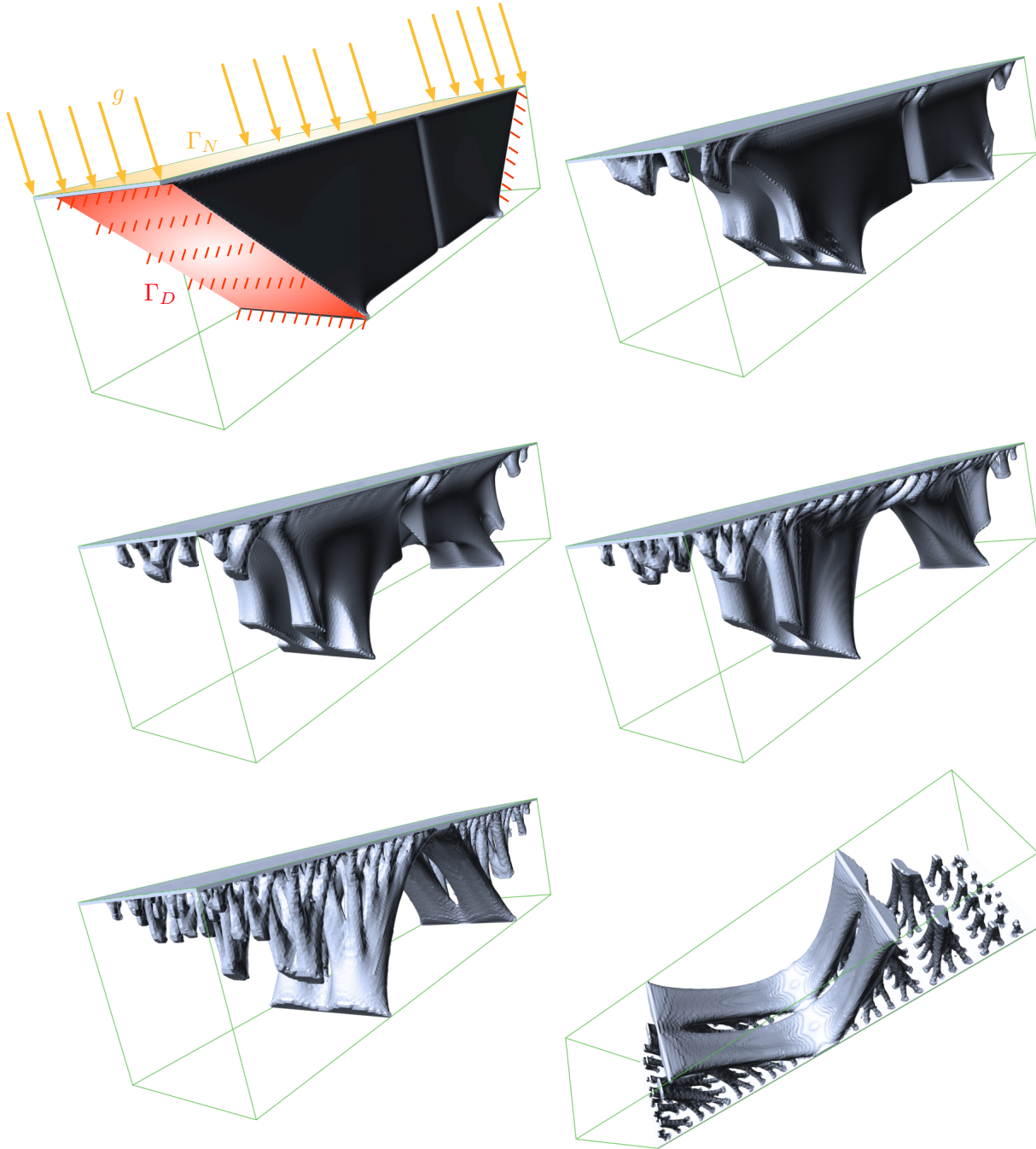


FIGURE 23. (From left to right, top to bottom) Optimization of the shape of a three-dimensional bridge: initial guess with details of the test case, iterations 15, 25, 40 and (bottom row) 200 (final shape).

see [121][158]. Note that, contrary to the compliance (2.7) used in all the previous numerical examples, the shape derivative of this functional features a non trivial adjoint state p_Ω (i.e. which is not a multiple of u_Ω). More precisely, a calculation based on the methods of Section 4 yields:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} Ae(u_\Omega) : e(p_\Omega) ds,$$

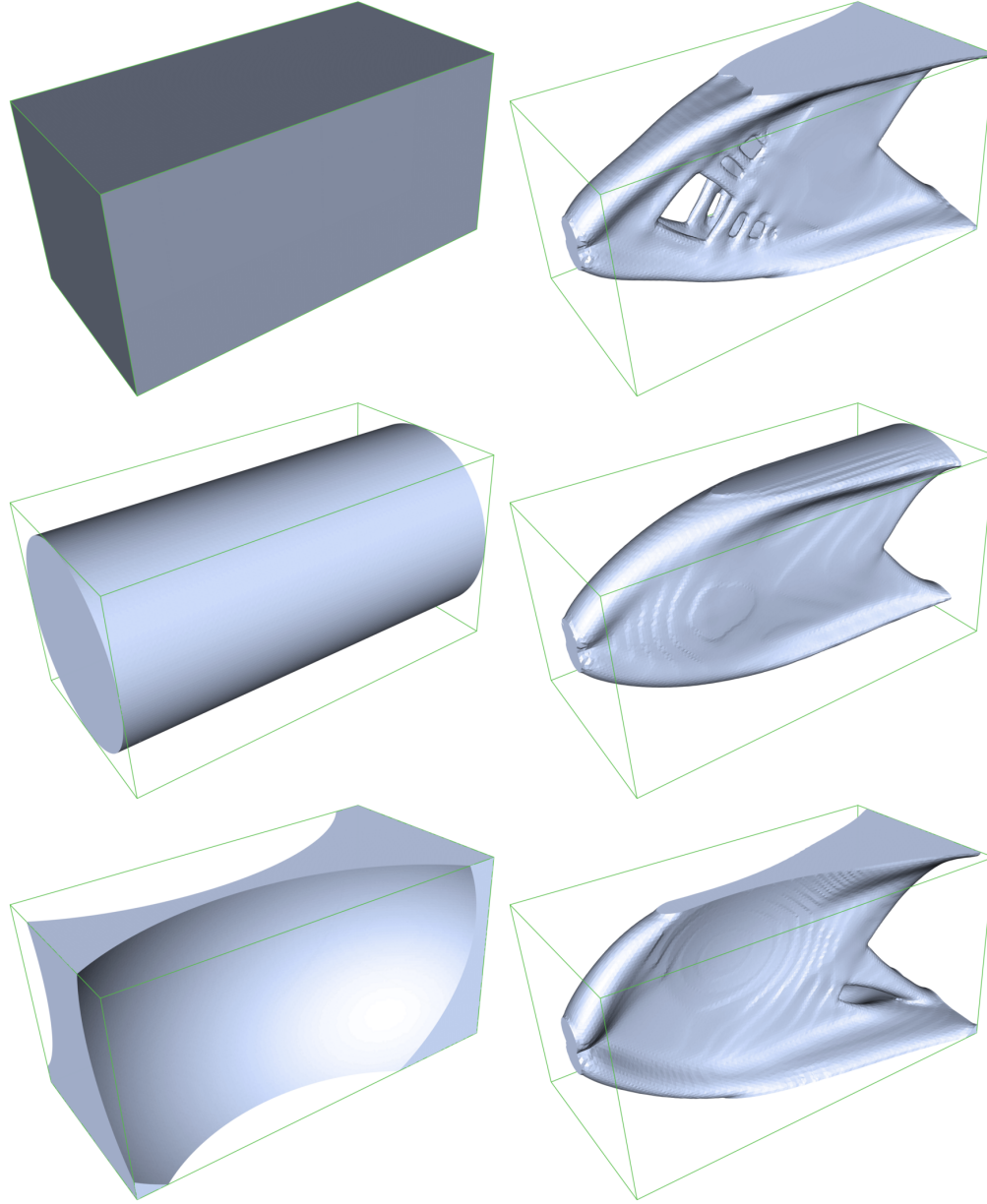


FIGURE 24. *Different optimization domains D (left column) leading to quite different optimal structures (right column) for the same forces, boundary conditions and imposed volume target $\text{Vol}(\Omega) = 0.15|\tilde{D}|$.*

where the adjoint state p_Ω is characterized by the system:

$$\begin{cases} -\text{div}(Ae(p_\Omega)) = \frac{2}{\int_{\omega_{\text{in}}} |u_d(x)|^2 dx} (u_\Omega \cdot d) & \text{in } \Omega, \\ p_\Omega = 0 & \text{on } \omega_{\text{in}}, \\ Ae(p_\Omega) = 0 & \text{on } \partial\Omega. \end{cases}$$

Taking advantage of the symmetry of the problem, only a quarter of the domain D is considered during the optimization, which is meshed with $40 \times 40 \times 80 = 128000$ hexaedral elements. The optimized shape is displayed in Figure 26 (bottom) in both reference and deformed configurations.

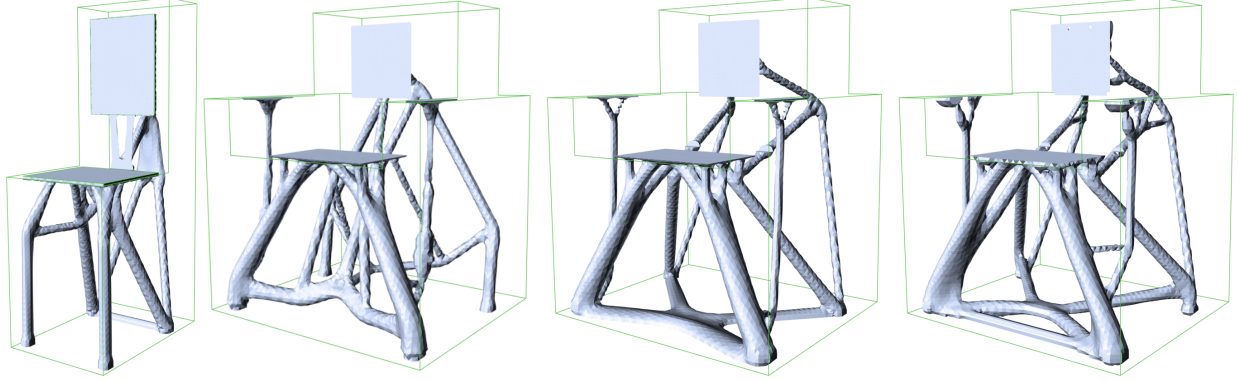


FIGURE 25. Various optimal chairs obtained from a full domain initialization. The final topology can be quite complex in 3d without any use of topological gradient.

7.5. Coupling the level set shape optimization method with the topological derivative

Both shape optimization approaches considered in Section 6 and 7 show weaknesses when the evolution of the shape Ω features topological changes. The geometric representation of Section 6 is downright incapable of dealing with the merger of separate holes inside Ω , since this would cause the mesh to become invalid. The level set shape optimization method of Section 7 is less restrictive in this regard: we have indeed seen that, up to a slight abuse of the theoretical framework, the implicit representation of Ω leaves the room for holes, or more generally distant regions of the shape boundary $\partial\Omega$ to moving closer and merge. However, there is no reason for holes to appear spontaneously inside the bulk of the shape; hence, if the topology of the initial design is too simple (or if the optimization path tends to make holes merge prematurely), the optimization process will result in a “poor” local minimum. Notice that this drawback is much more serious in 2d than in 3d where there are many more possibilities for remote regions of $\partial\Omega$ to collapse and create complex topologies from simple ones; see e.g. Figures 23, 25, 26.

In any event, it is of great interest to combine the information contained in shape derivatives with a mechanism to nucleate holes inside Ω . This is achieved by relying on the concept of *topological derivative*, that we now briefly discuss. The initial idea goes back to [91] but the first rigorous calculations of topological derivatives were introduced in [170, 171, 103]. We refer to [27, 54] for their introduction in the field of inverse problems, and to [15] for their coupling with shape derivatives in the context of structural optimization.

Topological derivatives rely on variations $\Omega_{x,r}$ of a shape Ω where a small hole with radius r is nucleated around $x \in \Omega$:

$$\Omega_{x,r} := \Omega \setminus \overline{B(x,r)}.$$

Typically, $B(x,r)$ is the open ball of center x and radius r but other shapes can also be considered. Note that in the partial differential equation defining the physical state $u_{\Omega_{x,r}}$, corresponding to the perturbed domain $\Omega_{x,r}$, a choice has to be made about the boundary conditions imposed on the boundary $\partial B(x,r)$ of the nucleated hole. Here, we restrict ourselves to homogeneous Neumann boundary conditions on $\partial B(x,r)$, which is the right setting for almost all the situations discussed in this chapter.

Definition 7.1. A function $J(\Omega)$, defined on open sets $\Omega \subset \mathbb{R}^d$, has a topological derivative $D_T J(\Omega)(x)$ at some point $x \in \Omega$ if the following expansion holds for small $r > 0$:

$$(7.14) \quad J(\Omega_{x,r}) = J(\Omega) + r^d D_T J(\Omega)(x) + o(r^d).$$

The calculation of topological asymptotic expansions is usually a delicate matter. It relies on a matched asymptotic expansion of the state $u_{\Omega_{x,r}}$ which, for small $r > 0$, is similar to u_Ω perturbed by a rescaled solution of some far-field problem around the unit obstacle $B(0,1)$. We shall not enter in any details and we refer to the book [144] for a complete discussion about topological derivatives. Rather, we limit ourselves with providing a few examples:

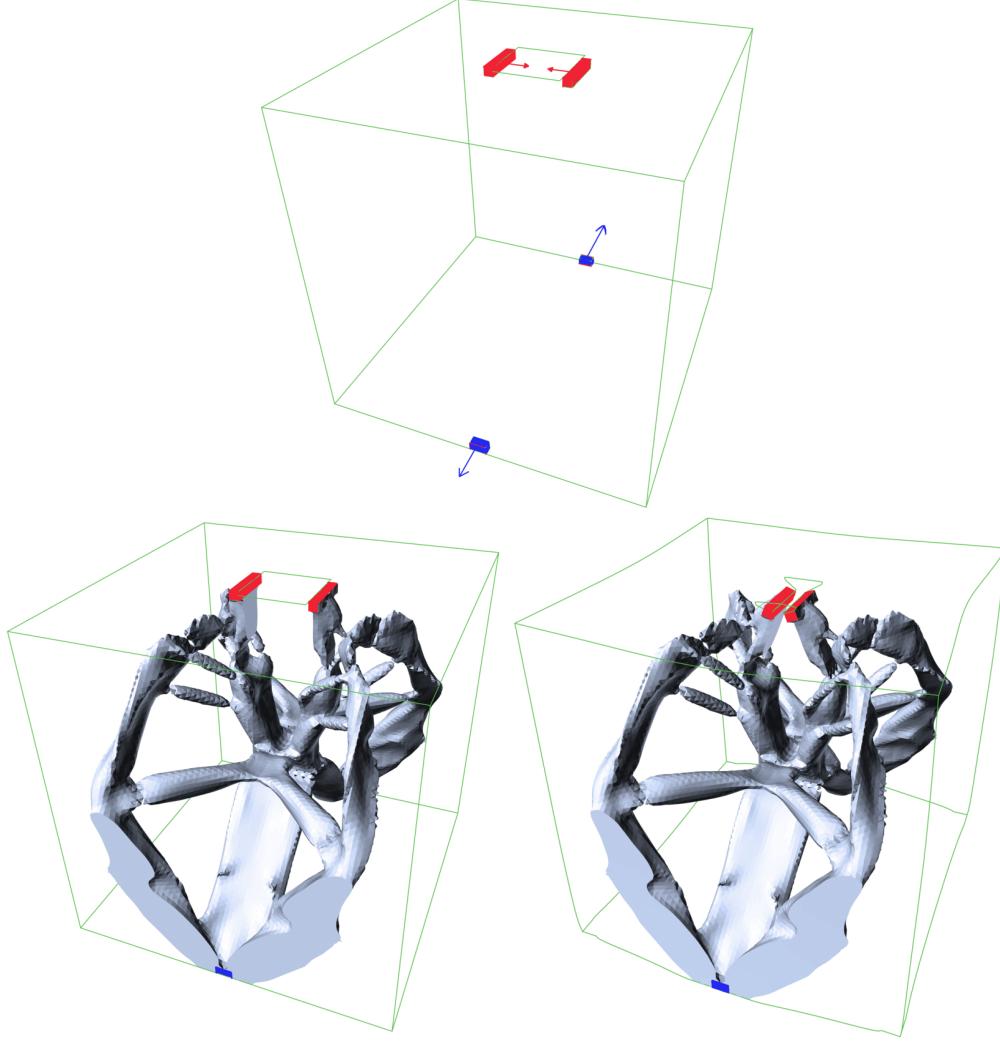


FIGURE 26. Optimization of a 3d gripping mechanism in Section 7.4.3; (top) setting of the test-case; (bottom, left) optimized shape and (bottom, right) deformed configuration: the red jaws tighten when the two symmetric blue zones on the bottom side are pulled.

- The volume $\text{Vol}(\Omega)$ has a topological derivative at every point $x \in \Omega$:

$$D_T \text{Vol}(\Omega)(x) = -\omega_d, \text{ where } \omega_d \text{ is the volume of the unit ball.}$$

- When $J(\Omega)$ is the compliance, given by (2.7) in the elasticity context, the topological derivative $D_T J(\Omega)(x)$ at every point $x \in \Omega$ is given, for $d = 2$, by:

$$D_T J(\Omega)(x) = \frac{\pi(\lambda + 2\mu)}{2\mu(\lambda + \mu)} \left(4\mu A e(u_\Omega) : e(u_\Omega) + (\lambda - \mu) \text{tr}(A e(u_\Omega)) \text{tr}(e(u_\Omega)) \right)(x).$$

and for $d = 3$:

$$D_T J(\Omega)(x) = \frac{\pi(\lambda + 2\mu)}{\mu(9\lambda + 14\mu)} \left(20\mu A e(u_\Omega) : e(u_\Omega) + (3\lambda - 2\mu) \text{tr}(A e(u_\Omega)) \text{tr}(e(u_\Omega)) \right)(x).$$

Topological derivatives find a natural role within the level set shape and topology optimization framework. For instance, every, say, 5 iterations in Algorithm 5 one can replace a standard level set iteration by a topological derivative iteration, as described in Algorithm 6 (where we omit for simplicity the superscript n referring to the current iteration).

Algorithm 6 Use of the topological derivative at a given stage of Algorithm 5.

Initialization: Shape Ω , defined via a level set function ϕ at the vertices of the mesh \mathcal{T} of D .

- (1) Calculate the state u_Ω and the adjoint p_Ω (if needed) on the mesh \mathcal{T} of D .
- (2) Calculate the quantity $D_T J(\Omega)(x)$ at every point $x \in \Omega$.
- (3) Identify the vertex $x \in \Omega$ of \mathcal{T} where $D_T J(\Omega)(x)$ is the most negative.

return The new shape $\Omega \setminus \overline{B(x, r)}$, where $r > 0$ is “small enough”.

A numerical illustration of this strategy is presented in Figure 27 where we come back to the setting of the 2d cantilever of Section 7.4.1, except that, now, the initial shape Ω^0 is the total computational domain D . The optimization is conducted by performing alternatively and periodically 5 iterations of the level set shape optimization Algorithm 5, followed by 1 iteration of Algorithm 6 to insert holes inside the shape Ω . The final design is very close to that of Figure 18. As a matter of fact, after iteration 50, the topological derivative of the minimized functional is positive everywhere inside D , so that no further hole insertion is needed from this point.

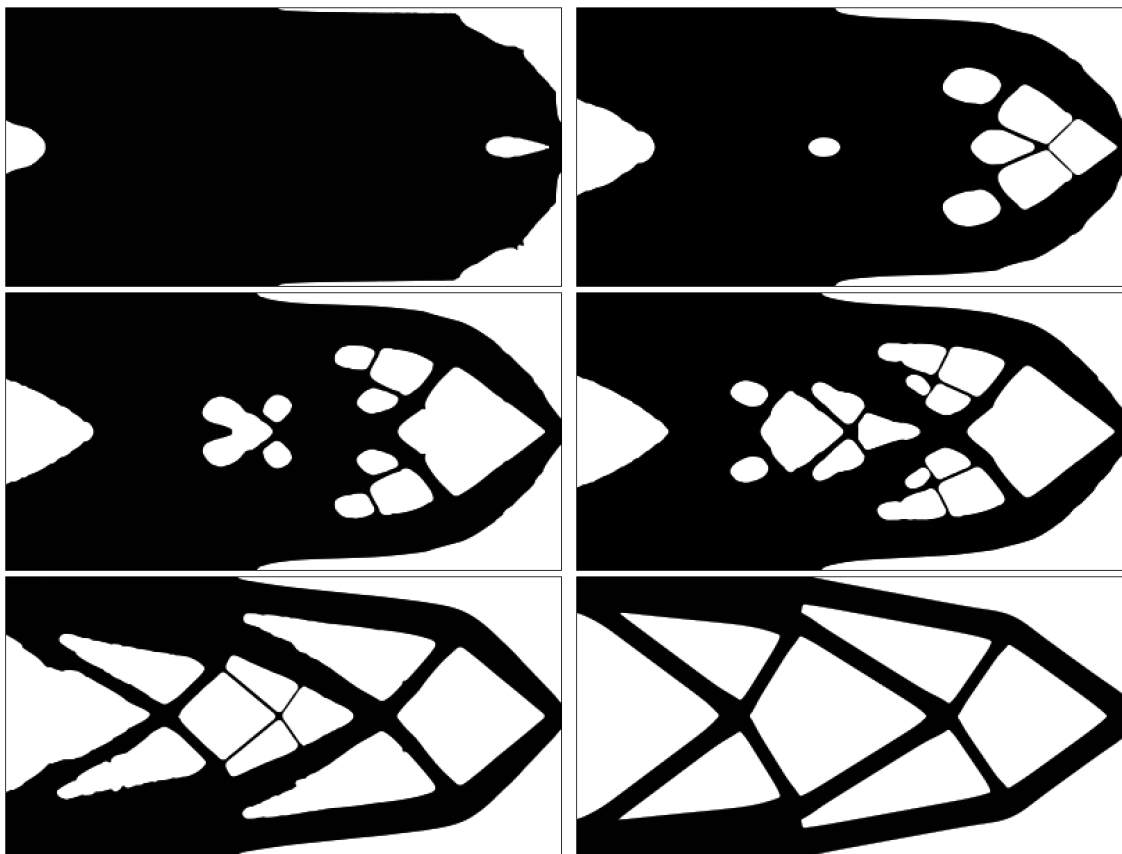


FIGURE 27. (From left to right, top to bottom) Iterations 15, 25, 30, 35, 50 and 200 in the 2d cantilever examples of Section 7.5, with the total domain D as initial shape and periodic use of the topological derivative.

Remark 7.6. The topological derivative is also used as the sole sensitivity indicator for functions of the domain in specific topology optimization algorithms (i.e. it is not coupled with shape derivatives); see [29] and Section 10.4 below for a short description.

7.6. Achievements of the level set method and a few numerical difficulties

We conclude this section devoted to the “classical” level set strategy for shape and topology optimization by taking stock of its main assets and drawbacks.

The foregoing numerical examples have already illustrated the most appealing features of the method:

- *Efficiency and robustness:* The level set method accounts for dramatic deformations of the shape in the course of the optimization process in a very robust way; in particular, topological changes are naturally handled.
- *Versatility:* Using the level set method to represent shapes and their deformations allows to decouple completely the mathematical formulation of the shape optimization problem and its numerical treatment.
- *Ease and efficiency of implementation:* The level set method is reasonably simple to implement; moreover, efficient open-source libraries are available to deal with the main operations involved in this framework, and notably the resolution of the level set evolution equations (7.2) and (7.3), or the redistancing procedure; see the references in Section 7.2. Besides, the additional CPU time required by these operations is totally negligible with respect to the most time-consuming operation of the shape optimization workflow, namely, the finite element resolution of the physical equations.

Aside from these assets, the level set method suffers from several difficulties:

- *Existence of multiple local minimizers and dependence on the initial guess:* The final design is quite dependent on the initial guess, and on the details of the practical implementation of the shape optimization process (the nature of the employed constrained optimization algorithm, the mesh size, etc.). These issues can be mitigated up to a certain point by relying on a number of strategies, such as the use of topological derivatives in 2d, or the addition of a perimeter constraint to reduce the sensitivity with respect to the mesh size. Note that these issues arise from the existence of many local minima for “generic” shape optimization problems, and from the homogenization effect which favors thinner and thinner patterns in the optimized shape; hence, they are common to many shape and topology optimization methods.
- *Need to approximate the state equation:* As we have already discussed in Remark 7.5, the used of a fixed mesh of the computational domain, where the shape is implicitly described, inevitably raises the need to approximate the physical equation for the state u_Ω ; this is, for instance, achieved by the ersatz material method in the context of linearized elasticity. Such an approximation could be too coarse, or difficult to devise in intricate physical contexts such as that of fluid-structure interacting systems; see Section 9.1.2. A more precise resolution of the state equation can be achieved by using X-FEM, or cutfem methods, but these methods are *intrusive*: the finite element solver cannot be used as a “black-box”, which is often a strong requirement in the industrial context. This point motivates the level set based mesh evolution method that we next present in Section 8.

8. A LAGRANGIAN, LEVEL SET BASED MESH EVOLUTION METHOD FOR SHAPE AND TOPOLOGY OPTIMIZATION

We describe in this section how the “classical” level set shape and topology optimization method of Section 7.3 can be enriched so that an exact mesh of each shape Ω^n , produced in the course of the optimization process, be available for finite element computations. This method was introduced in [13], then systematized in [14]; recent developments, including applications to large-scale multidisciplinary problems are presented in [95, 96].

8.1. Presentation of the method

Let us consider, once again, the numerical resolution of the unconstrained shape optimization problem (2.9) in the context of linear elastic structures presented in Section 2.1.1. From the “classical” level set shape optimization strategy in Section 7, we retain the idea that the evolving shape Ω (standing for any of the iterates Ω^n , $n \geq 0$) should be consistently contained in a fixed computational domain D . In the present context, however, D is equipped with a conforming, simplicial mesh \mathcal{T} , with high quality for finite element calculations; see Section 6.1 for definitions. Contrary to the “classical” level set shape optimization method of Section 7, the mesh \mathcal{T} is no longer fixed; the novel ingredient is that it is consistently modified, in such

as way that Ω explicitly appears as a submesh \mathcal{S} of \mathcal{T} . Hence, two complementary representations of Ω are now available at all stages of the optimization workflow; see Figure 28 for an illustration.

- A *level set representation*: Ω is the negative subdomain of a level set function $\phi : D \rightarrow \mathbb{R}$, defined at the vertices of \mathcal{T} . As we have seen in Section 7.1, this description is very well-suited to account for the deformation of Ω according to a prescribed vector field θ .
- A *meshed representation*: Ω is equipped with a simplicial mesh \mathcal{S} , which is a *submesh* of that \mathcal{T} of D . Under this form, it is simple to perform the mechanical calculation of u_Ω (or p_Ω) on Ω via the finite element method.

A typical implementation of this strategy is summarized in Algorithm 7.

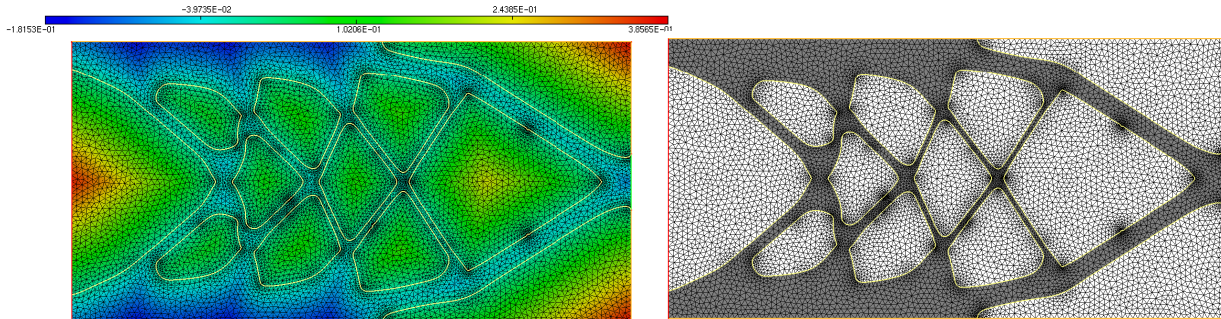


FIGURE 28. A shape $\Omega \subset D$ can be described (left) by a level set function ϕ defined at the vertices of the mesh \mathcal{T} of D , or (right) as the submesh \mathcal{S} .

It combines the assets of both representations of the optimized shape Ω . On the one hand, the meshed representation of Ω allows to perform the finite element calculations needed in the resolution of the state and adjoint equations for u_Ω and p_Ω without any approximation. In particular, the ersatz material approximation is not needed. As we have hinted at in Section 7.6, this fact is particularly crucial in applications where such approximations would be difficult to devise, or perhaps too inaccurate, as in the context of fluid-structure interacting systems (see Section 9.1.2). Besides, that an exact, high-quality mesh of Ω be available at each stage of the optimization process makes this method *non intrusive*: any external solver could be used as a black box. On the other hand, since the update of the shape between iterations is accounted for by the level set method, dramatic deformations are possible, including topological changes.

As we have seen, the cornerstone of this strategy is a set of numerical methods for switching from a meshed representation of a shape $\Omega \subset D$ to its level set representation, and conversely. We next discuss these operations, corresponding to the Steps 1 and 5 in Algorithm 7.

8.2. Switching between the level set and meshed descriptions of shapes

The practical implementation of Algorithm 7 requires at first a numerical method for generating a level set function ϕ for a shape Ω at the vertices of the simplicial mesh \mathcal{T} of D (see Step 1 in Algorithm 7). As we have seen in Section 7.2.3, it is often convenient to calculate the signed distance function d_Ω to Ω for this purpose, and this can be realized by using the numerical methods evoked in there.

The converse operation (Step 5 in Algorithm 7) is much more delicate and not classical in the literature: it consists in passing from a level set representation of a shape Ω to its meshed representation. More precisely, let $\phi : D \rightarrow \mathbb{R}$ be a level set function, given at the vertices of a simplicial mesh \mathcal{T} of D , defining a shape $\Omega \subset D$ in the sense that (7.1) holds; see Figure 29 (a). We aim to create a new mesh \mathcal{T}^* of D where Ω explicitly appears as a submesh. To achieve this, we proceed within two steps, which are illustrated on Figure 29:

- (1) The 0 level set of ϕ is identified by linear interpolation inside the simplices of \mathcal{T} , from the values at its vertices. This 0 level set is enforced in the mesh by splitting explicitly the simplices crossed by this isoline. This simple and combinatorial operation results in a new mesh $\mathcal{T}_{\text{temp}}$ of D , which is

Algorithm 7 Level set based mesh evolution method for problem (2.9).

Initialization: The computational domain D is equipped with a conforming mesh \mathcal{T}^0 , containing a submesh \mathcal{S}^0 of the initial shape Ω^0 .

for $n = 0, \dots$, until convergence **do**

- (1) Generate a level set function ϕ^n for Ω^n at the vertices of \mathcal{T}^n .
- (2) Calculate on the submesh \mathcal{S}^n the solution u_{Ω^n} to the exact state equation (2.1), and that p_{Ω^n} to the adjoint equation.
- (3) Use the derivative $J'(\Omega^n)(\theta)$ (see Section 4) to infer a (scalar) descent direction for $J(\Omega)$ by solving the (scalar) extension-regularization problem on the whole (mesh of) D :

$$\forall v \in H, \quad a(v_{\Omega^n}, v) = -J'(\Omega^n)(v n_{\Omega^n}),$$

where for instance:

$$H = H_{\Gamma_D \cup \Gamma_N}^1(D), \quad \text{and} \quad a(u, v) = \alpha^2 \int_D \nabla u \cdot \nabla v \, dx + \int_D uv \, dx.$$

- (4) Solve the level set Hamilton-Jacobi equation:

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + v_{\Omega^n}(x) |\nabla \phi(t, x)| = 0 & \text{for } t \in (0, \tau^n), \quad x \in D, \\ \phi(0, x) = \phi^n(x) & \text{for } x \in D, \end{cases}$$

on the unstructured mesh \mathcal{T}^n of D , over a suitable time period $(0, \tau^n)$.

- (5) Discretize the resulting level set function $\phi(\tau^n, \cdot)$ into the mesh \mathcal{T}^n to obtain a new mesh \mathcal{T}^{n+1} of D where Ω^{n+1} is explicitly discretized by a submesh \mathcal{S}^{n+1} .

end for

return Ω^n

conforming, and where Ω explicitly appears as a submesh. The mesh $\mathcal{T}_{\text{temp}}$ has unfortunately poor element quality: as the intersection pattern of $\partial\Omega$ with \mathcal{T} is not controlled, very stretched, nearly flat elements are created in the process; see Figure 29 (b).

- (2) Using the local remeshing operations described in Section 6.2.1, the quality of $\mathcal{T}_{\text{temp}}$ is gradually improved, so that the resulting mesh \mathcal{T}^* has fine element quality; see Figure 29 (c).

These features are implemented in the open-source remeshing library `mmg` [73]; see [72] for the underlying theoretical background.

8.3. Numerical examples

We finally present three numerical examples where shape and topology optimization problems are addressed with the level set based mesh evolution method of this section.

The first one is, again, the 2d benchmark cantilever test-case, in the exact same setting as in Section 7.4.1. Several intermediate shapes arising in the course of the evolution are reported in Figure 30, where the submesh \mathcal{S}^n is clearly distinguishable inside the mesh \mathcal{T}^n . Each mesh \mathcal{T}^n contains about 11,000 vertices, for about twice as many triangles.

We next go over the three-dimensional cantilever test-case introduced in Section 7.4.3; see Figure 31 for some snapshots of the optimization process (only the skin of the submesh \mathcal{S}^n is plotted). Each mesh \mathcal{T}^n contains about 32,000 vertices, for about 6 times as many tetrahedra.

Finally, we reconsider the shape optimization problem of a 3d bridge from Section 7.4.3, whose results are displayed in Figure 32. Each mesh \mathcal{T}^n contains about 35,000 vertices, for about 6 times as many tetrahedra.

Interestingly, in all three cases, the optimized shapes are quite similar to those obtained in Section 7, by using the “classical” level set method for shape and topology optimization. The shape Ω experiences dramatic changes in the course of the evolution; nevertheless, an exact, high quality mesh of Ω is available at each iteration, which enables a precise resolution of the physical equation (2.1) by the finite element method (in particular, the ersatz material approximation is not needed).

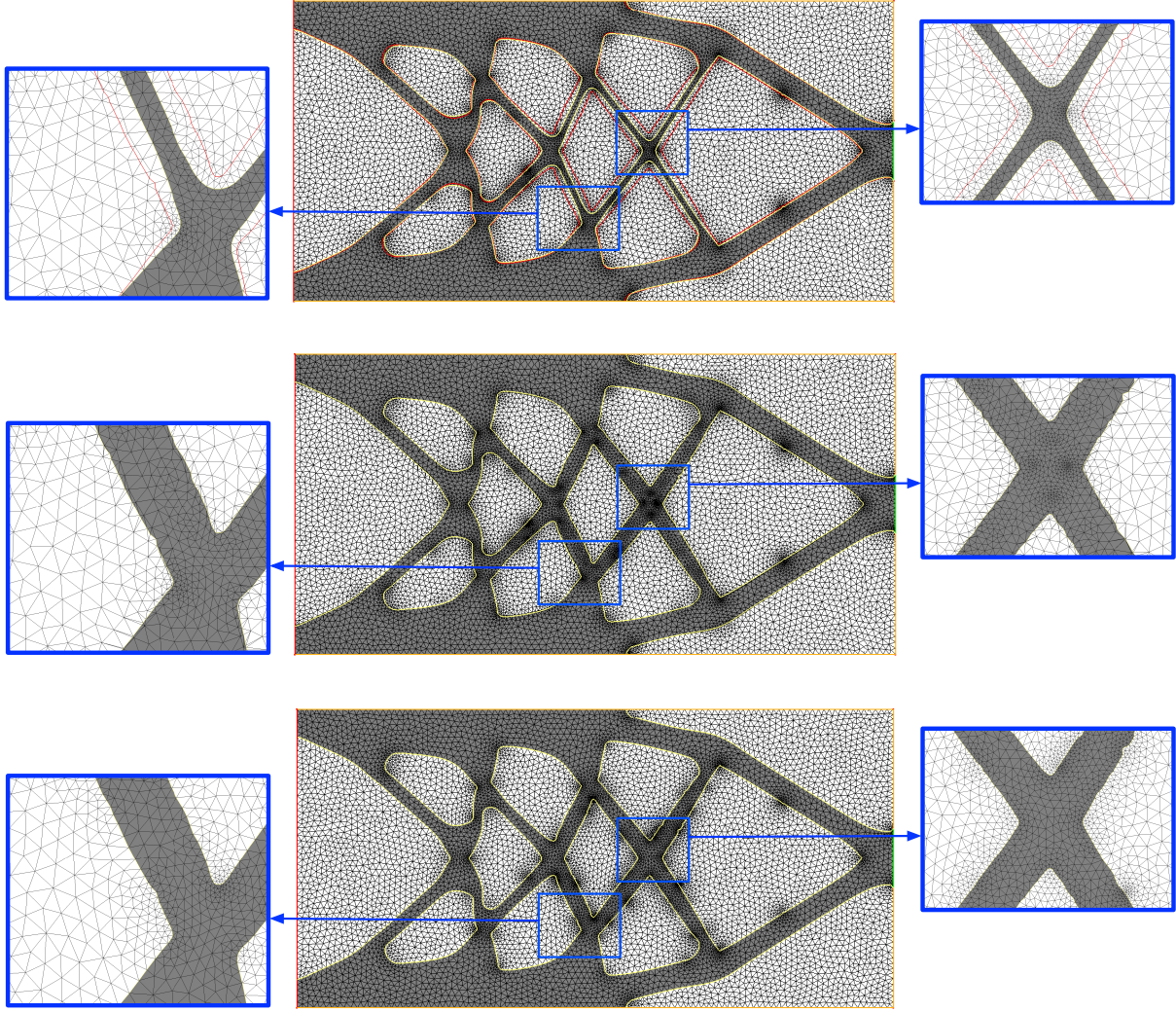


FIGURE 29. (Top) 0 level set (in red) of the function ϕ defined at the vertices of the mesh \mathcal{T} ; (Middle) Intermediate mesh $\mathcal{T}_{\text{temp}}$ with poor quality; (bottom) high-quality mesh \mathcal{T}^* of D , featuring a submesh of the new shape.

9. SHAPE OPTIMIZATION IN DIFFERENT PHYSICAL CONTEXTS AND A FEW RECENT CHALLENGES

In this section, we present some natural applications and extensions of the level set shape and topology optimization framework discussed in this chapter. In Section 9.1, we exemplify applications of our level set based algorithms to the optimization of more intricate physical objects than mechanical structures, namely nanophotonics devices and fluid-structure systems. The next Section 9.2 is dedicated to the important topic of geometric constraints, imposed by the manufacturing process: a quite general model, based on the notion of signed distance function, is presented for such constraints, which conveniently slots into the level set framework.

9.1. Beyond linearized elasticity: applications to different physical contexts

We present in this section two examples arising from other physical fields than those of thermal or structural mechanics. These demonstrate the versatility of the shape and topology optimization techniques discussed in this chapter.

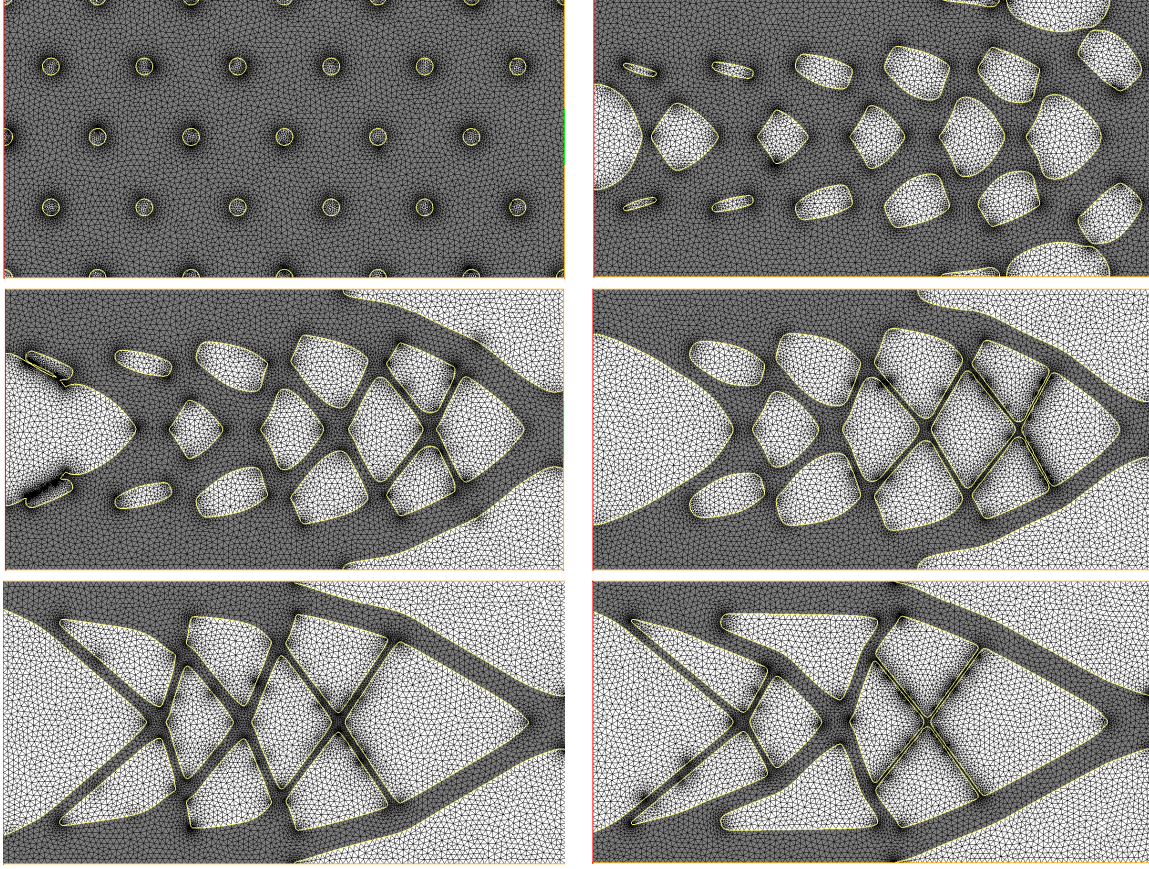


FIGURE 30. (From left to right, top to bottom) Iterations 0, 20, 30, 40, 100 and 300 of the 2d cantilever example tackled with the level set mesh evolution algorithm of Section 8.

9.1.1. Shape optimization of nanophotonic devices

Nanophotonic devices are components of photonic integrated circuits allowing for the manipulation of light, described as an electromagnetic wave, at the nanometric scale. Their fascinating properties are obtained by optimizing the repartition of a *core* and a *cladding* material; see [93] for an overview.

Let us consider the physical setting depicted in Figure 33: the component is sought as an arrangement of a matrix Ω of Silicium Si and cladding $D \setminus \bar{\Omega}$ by air, within a fixed design domain $D \subset \mathbb{R}^3$. This design domain D is mounted on an SiO_2 substrate; it is connected via the region $\Gamma_{\text{in}} \subset \partial D$ to one (or several) input waveguide, where light is injected as a guided mode, and via Γ_{out} to one (or several) output waveguide, transmitting light to the other components of the optical circuit.

A typical, multi-purpose shape optimization problem in this context consists in maximizing the power carried by the i^{th} guided mode of the output waveguide, characterized by its electric and magnetic fields E_i , H_i :

$$(9.1) \quad \max_{\Omega \in \mathcal{U}_{\text{ad}}} \left| \frac{1}{2} \int_{\Gamma_{\text{out}}} E_{\Omega} \times H_i \, ds \right|^2.$$

The admissible shapes $\Omega \in \mathcal{U}_{\text{ad}}$ are imposed to be invariant in the vertical direction. The electric field E_{Ω} in (9.1) is the solution to the time-harmonic Maxwell equation at given frequency ω , wavelength $\lambda = \frac{2\pi c}{\omega}$ and associated wavenumber $k = \frac{2\pi}{\lambda}$:

$$\nabla \times \nabla \times E_{\Omega} - k^2 n_{\Omega}^2 E_{\Omega} = 0,$$

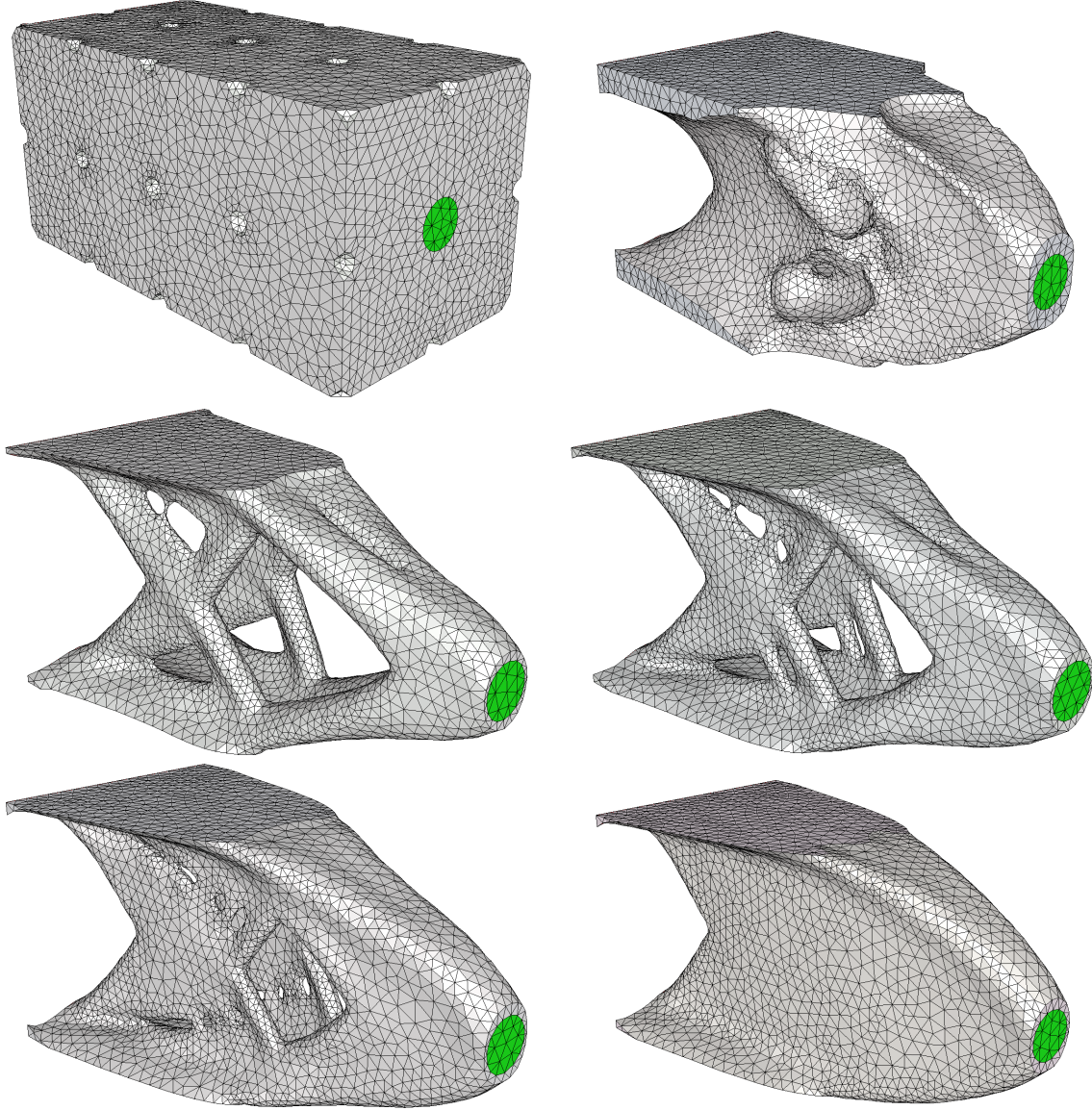


FIGURE 31. (From left to right, top to bottom) Iterations 0, 27, 45, 60, 80 and 200 of the 3d cantilever example tackled with the level set mesh evolution algorithm of Section 8.

where the optical index is

$$n_{\Omega}(x) = \begin{cases} n_{\text{Si}} & \text{if } x \in \Omega, \\ n_{\text{air}} & \text{if } x \in D \setminus \Omega. \end{cases}$$

This system is complemented with light injection boundary conditions on Γ_{in} , and with the Silver-Müller radiation condition at infinity.

A numerical example of a shape and topology optimization analysis conducted in this setting is presented in Figure 34, which is excerpted from the article [128]. The optimized component is a *diplexer*: it operates under two different wavelengths $\lambda_1 = 1.55\mu\text{m}$, $\lambda_2 = 1.31\mu\text{m}$. In both cases, light is injected as the fundamental electric mode E_0 of the input waveguide, and the component is optimized so that the largest possible fraction of the incoming light is conveyed to the left-hand output waveguide when the operating wavelength equals λ_1 , and to the right-hand output waveguide when it equals λ_2 .

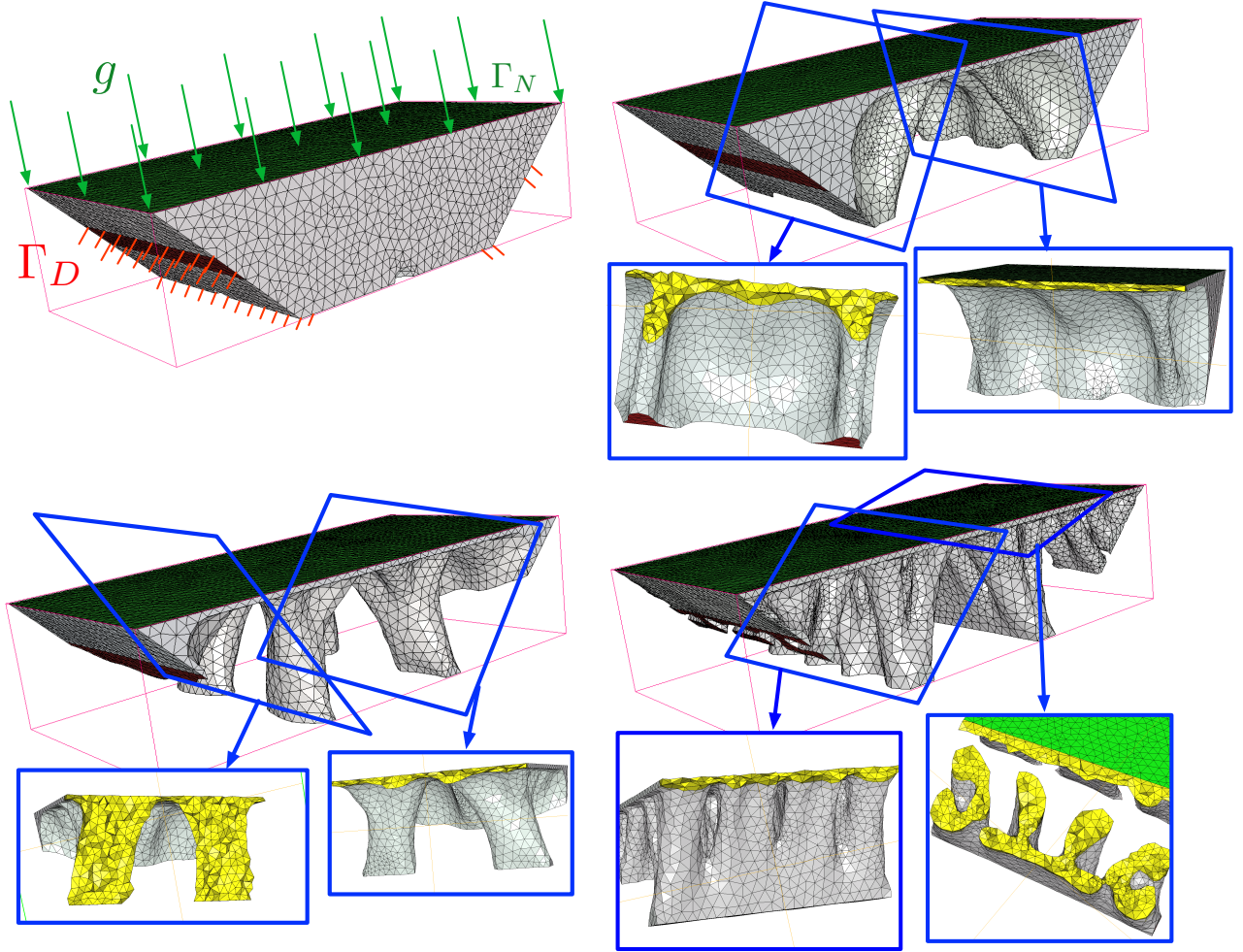


FIGURE 32. (From left to right, top to bottom) Iterations 0 (with details of the test case), 50, 70 and 200 of the 3d bridge example tackled with the level set mesh evolution algorithm of Section 8.

9.1.2. Optimization of the design of coupled fluid-structure systems

We now turn to an optimization example of the interface between a structure and a surrounding fluid. This example is taken from [96]; see also [95] for more details. A cavity $D \subset \mathbb{R}^3$ is divided into a solid obstacle $\Omega_s = \Omega$ and a fluid phase $\Omega_f := D \setminus \overline{\Omega}$ with viscosity ν and density ρ , entering D from the side Γ_{in} , with given velocity v_0 . We denote by $\Gamma = \partial\Omega_f \cap \partial\Omega_s$ the interface between fluid and solid phases; see Figure 35 for an illustration.

The fluid contained in Ω_f is described by its velocity v_Ω and pressure p_Ω , which are solution to the steady-state Navier-Stokes equations:

$$(9.2) \quad \begin{cases} -\operatorname{div}(\sigma_f(v_\Omega, p_\Omega)) + (v_\Omega \cdot \nabla)v_\Omega + \nabla p_\Omega = f & \text{in } \Omega_f, \\ \operatorname{div} v_\Omega = 0 & \text{in } \Omega_f, \\ v = 0 & \text{on } \Gamma, \\ v = v_0 & \text{on } \Gamma_{\text{in}}, \\ \sigma_f(v_\Omega, p_\Omega)n = 0 & \text{on } \partial\Omega_f \setminus (\Gamma_{\text{in}} \cup \Gamma), \end{cases}$$

where the body force f accounts for gravity effects, and $\sigma_f(v, p) = 2\nu e(v) - p\mathbf{I}$ is the fluid stress tensor.

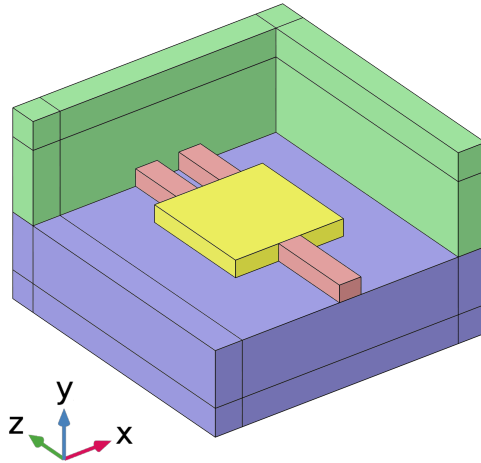


FIGURE 33. Setting of the nanophotonic device optimization example of Section 9.1.1. The SiO_2 substrate is depicted in blue, the input and output waveguides are in red, the design domain is in yellow, and the air cladding is in green.

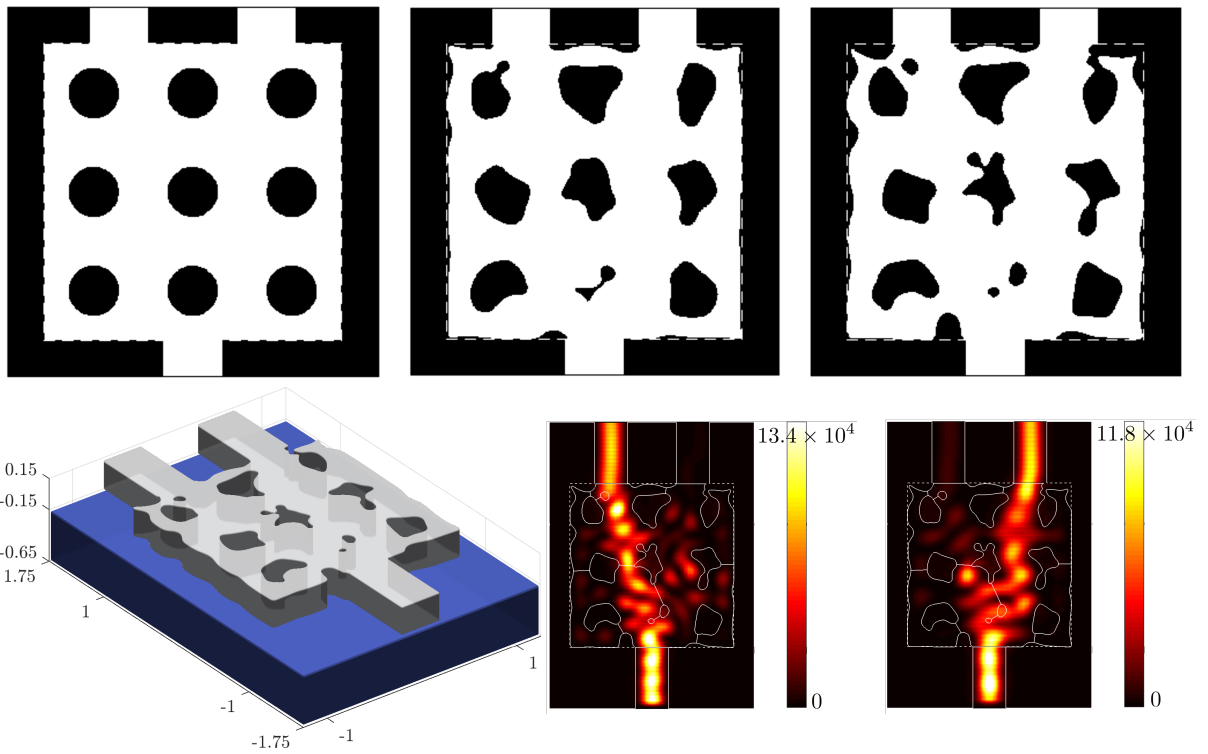


FIGURE 34. (Upper row) Initial, intermediate, and final shapes in the optimization of a diplexer (the cuts in the horizontal (x, z) plane are displayed); (lower row) optimized shape, and energy densities at both wavelengths.

The solid obstacle Ω_s is made of a (non-optimizable) vertical bar, attached to the bottom side of ∂D , which has to be reinforced so that it better withstands the stress $\sigma_f(v_\Omega, p_\Omega)n$ imposed by the fluid. The

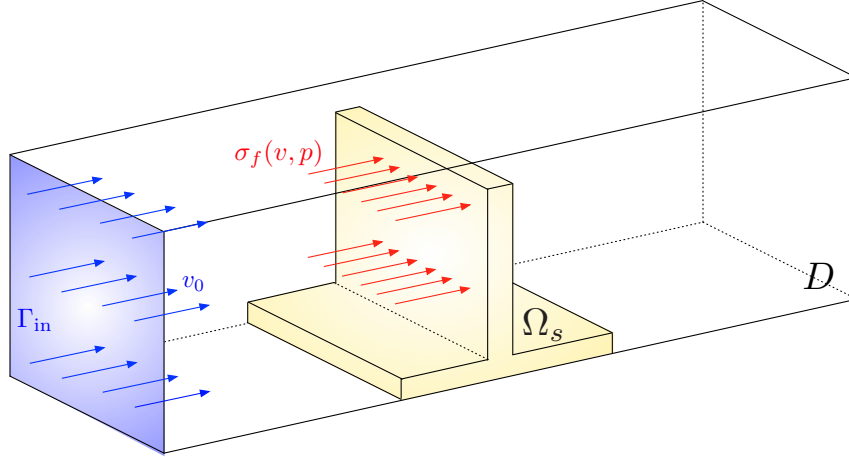


FIGURE 35. *Setting of the fluid-structure interaction problem of Section 9.1.2.*

elastic displacement $u_\Omega : \Omega_s \rightarrow \mathbb{R}^3$ of Ω_s is mathematically described as the solution to the elasticity equation:

$$(9.3) \quad \begin{cases} -\operatorname{div}(Ae(u_\Omega)) = 0 & \text{in } \Omega_s, \\ Ae(u_\Omega)n = \sigma_f(v_\Omega, p_\Omega)n & \text{on } \Gamma, \\ u_\Omega = 0 & \text{on } \partial\Omega_s \setminus \Gamma. \end{cases}$$

The shape optimization problem reads (with $\Omega_s = \Omega$):

$$(9.4) \quad \min_{\Omega \subset D} J(\Omega) \text{ s.t. } \operatorname{Vol}(\Omega) \leq V_T,$$

where V_T is a volume target and $J(\Omega)$ is the solid compliance

$$J(\Omega) := \int_{\Omega_s} Ae(u_\Omega) : e(u_\Omega) \, dx.$$

This problem features delicate physical equations. In particular, its treatment by any fixed mesh shape optimization method would demand an approximation of the difficult boundary conditions

$$v_\Omega = 0 \text{ and } Ae(u_\Omega)n = \sigma_f(v_\Omega, p_\Omega)n \text{ on } \Gamma.$$

To circumvent this difficulty, (9.4) is solved thanks to the level set based mesh evolution method presented in Section 8, which conveniently allows to solve accurately the coupled physical equations (9.2)-(9.3) on an exact mesh of Ω_f and Ω_s , without any approximation. Several intermediate designs are reported on Figure 36 where only the mesh of the fluid-solid interface Γ is plotted.

9.2. Geometric and manufacturing constraints

One main objection from engineers towards almost any shape and topology optimization technique is that the optimized designs are generally difficult, not to say impossible, to fabricate in practice. To circumvent this drawback, a large amount of investigations has been devoted to modelling constraints imposed on shapes by the manufacturing process; see [20, 112, 115, 134, 155, 167], to name a few contributions.

Often, these constraints are of a geometric nature: they are related to the thickness of shapes, to their radii of curvature, etc. In this section, we present a quite general and flexible formulation for such geometric constraints, whose implementation is considerably eased by the use of the level set method; see [20, 134] for more details.

The key ingredient of this framework is the signed distance function d_Ω to a shape Ω , defined in (7.2.3), which encodes all the geometric information that is needed about Ω . We collect some additional material about the signed distance function in Section 9.2.1, before exemplifying its role in the modelling of geometric constraints in Section 9.2.2, and providing numerical examples in Section 9.2.3.

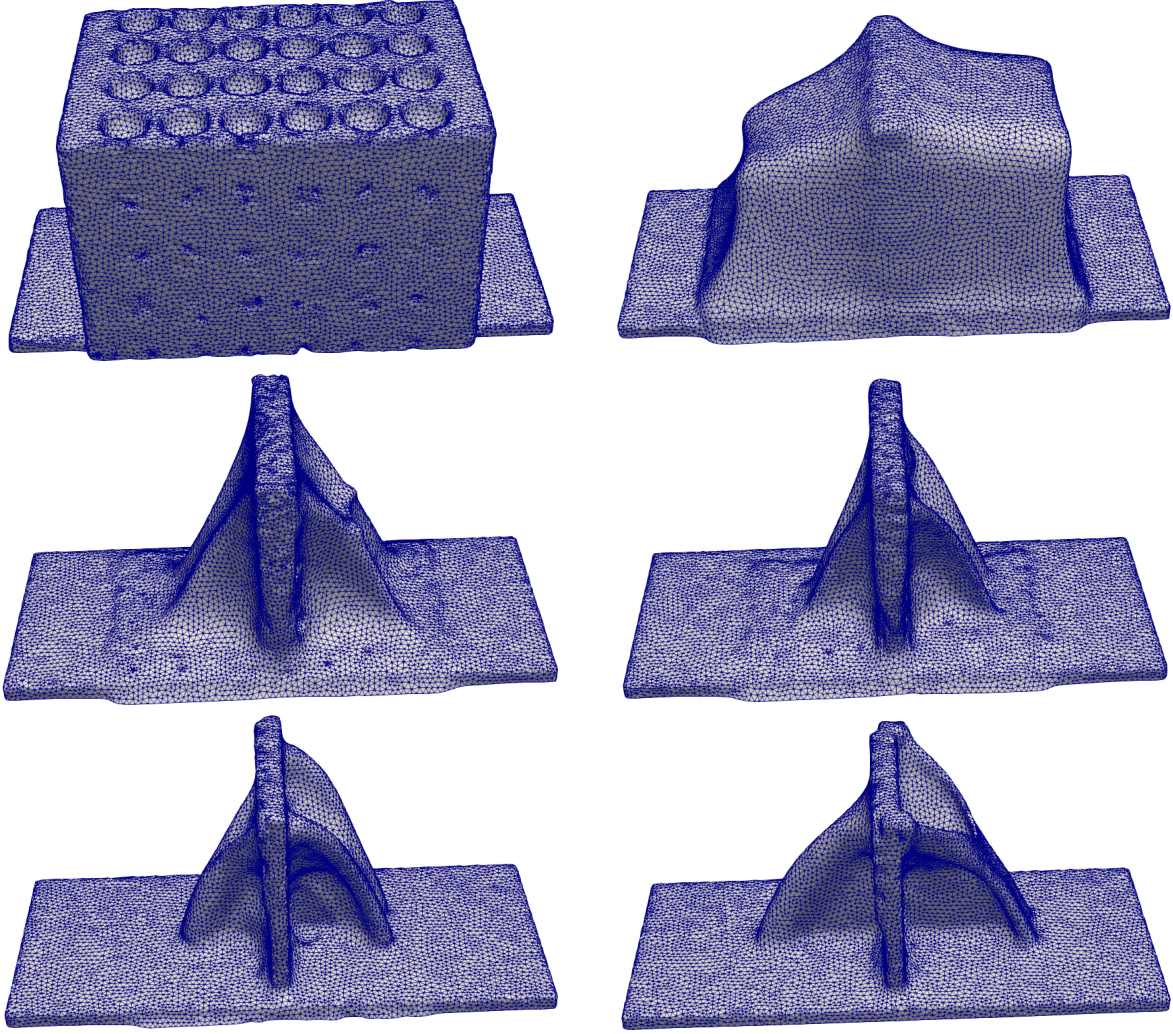


FIGURE 36. (From left to right, top to bottom) Iterations 0, 40, 100, 125, 175 and 300 of the fluid-structure interaction example of Section 9.1.2.

9.2.1. Further considerations about the signed distance function

Let us first introduce a few concepts related to the signed distance function; these are illustrated in Figure 37.

Definition 9.1. Let $\Omega \subset \mathbb{R}^d$ be a bounded, Lipschitz domain.

- For any point $x \in \mathbb{R}^d$, the set $\Pi_{\partial\Omega}(x)$ of projections of x onto $\partial\Omega$ is:

$$\Pi_{\partial\Omega}(x) = \{y \in \partial\Omega, |x - y| = d(x, \partial\Omega)\}.$$

When this set is a singleton, its unique element $p_{\partial\Omega}(x)$ is called the projection of x onto $\partial\Omega$.

- The skeleton Σ is the set of points $x \in \mathbb{R}^d$ having at least two projections onto $\partial\Omega$. Note that by definition, $\Sigma \cap \partial\Omega = \emptyset$.
- The ray emerging from $y \in \partial\Omega$ is the set of points in \mathbb{R}^d admitting y as unique projection:

$$(9.5) \quad \text{ray}_{\partial\Omega}(y) = \{x \in \mathbb{R}^d \setminus \Sigma, p_{\partial\Omega}(x) = y\}.$$

The following properties hold about these concepts; see [41, 82] for proofs.

Proposition 9.1.

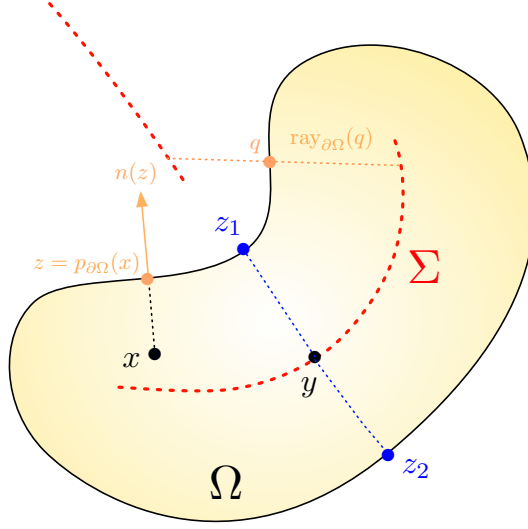


FIGURE 37. Illustration of Definition 9.1: the point x does not belong to the skeleton Σ , and has $z \in \partial\Omega$ as unique projection point; the point y belongs to Σ and $\Pi_{\partial\Omega}(y) = \{z_1, z_2\}$.

- One point $x \in \mathbb{R}^d$ belongs to the complement $\mathbb{R}^d \setminus \Sigma$ if and only if d_Ω^2 is differentiable at x . Since d_Ω is Lipschitz (and so is differentiable a.e. owing to Rademacher's theorem; see [92]), Σ has zero Lebesgue measure.
- If Ω is of class \mathcal{C}^1 and $x \notin \Sigma$, one has:

$$\nabla d_\Omega(x) = n(p_{\partial\Omega}(x)).$$

- If Ω is of class \mathcal{C}^k for some $k \geq 2$, there exists a neighborhood U of $\partial\Omega$ where d_Ω is of class \mathcal{C}^k .

Let us now turn to the differentiation of the signed distance function with respect to the domain. The first result of interest in this direction is related to the ‘‘Eulerian’’ derivative of the mapping $\Omega \mapsto d_\Omega$; we refer to [81] for the proof.

Proposition 9.2. *Let Ω be a bounded domain of class \mathcal{C}^1 , and let $x \in \mathbb{R}^d \setminus \Sigma$. Then the mapping $\Omega \mapsto d_\Omega(x)$ is differentiable with respect to the domain; the associated derivative $d'_\Omega(\theta)(x)$ reads:*

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad d'_\Omega(\theta)(x) = -\theta(p_{\partial\Omega}(x)) \cdot n(p_{\partial\Omega}(x)).$$

Let now D be a hold-all domain, and consider a function of the domain under the form:

$$(9.6) \quad J(\Omega) = \int_D j(x, d_\Omega(x)) \, dx,$$

where $j : \mathbb{R}_x \times \mathbb{R}_s \rightarrow \mathbb{R}$ is a smooth function.

Proposition 9.3. *The shape functional $J(\Omega)$ in (9.6); is shape differentiable at any bounded domain $\Omega \Subset D$ of class \mathcal{C}^1 , with derivative*

$$(9.7) \quad J'(\Omega)(\theta) = - \int_D \frac{\partial j}{\partial s}(x, d_\Omega(x)) \theta(p_{\partial\Omega}(x)) \cdot n(p_{\partial\Omega}(x)) \, dx.$$

If in addition, Ω is of class \mathcal{C}^3 , this rewrites:

$$(9.8) \quad J'(\Omega)(\theta) = - \int_{\partial\Omega} v_\Omega(y) \theta(y) \cdot n(y) \, ds(y), \quad \text{where}$$

$$v_\Omega(y) := \int_{\text{ray}_{\partial\Omega}(y) \cap D} \left(\prod_{i=1}^{d-1} (1 + d_\Omega(z) \kappa_i(y)) \right) \frac{\partial j}{\partial s}(z, d_\Omega(z)) \, dl(z).$$

In the above formula, the $\kappa_i(y)$, $i = 1, \dots, d-1$ are the principal curvatures of $\partial\Omega$ at y , and the rays $\text{ray}_{\partial\Omega}(y)$ are defined by (9.5).

Hint of proof. Formula (9.7) follows from the Lebesgue dominated convergence theorem, using Proposition 9.2, and the fact that Σ has null Lebesgue measure.

The derivation of the second formula (9.8) is more intricate, and we refer to [11]. Briefly, it stems from the coarea formula (a “curved” version of the Fubini theorem); see [92]. The assumption whereby Ω should have \mathcal{C}^3 regularity is needed so that not only Σ , but also $\bar{\Sigma}$ have null Lebesgue measure; see [58]. \square

9.2.2. Formulation of some geometric constraint functionals

A wide variety of manufacturing constraints can be modelled by functionals of the form (9.6) for some adequate, smooth function $j : \mathbb{R} \rightarrow \mathbb{R}$. For brevity, we limit ourselves in this direction with a presentation of the crucial thickness constraints:

- *Minimum thickness:* In practice, it is crucial that the produced designs do not exhibit very thin and frail parts, that would likely break during the machining or cooling. One possible mathematical characterization of a shape Ω having minimum thickness larger than d_{\min} is:

$$(9.9) \quad \forall x \in \partial\Omega, \quad \forall t \in [0, d_{\min}], \quad d_{\Omega}(x - tn(x)) \leq 0.$$

Such a requirement can be turned into an equality constraint functional of the form (9.6):

$$P_{\min T}(\Omega) = 0, \quad \text{where } P_{\min T}(\Omega) := \int_{\partial\Omega} \int_0^{d_{\min}} [d_{\Omega}(x - tn(x))]_+^2 dt ds(x),$$

where $[t]_+ = \max(t, 0)$ is the positive part of a real number t .

- *Maximum thickness:* Besides, it is often required that shapes do not present too thick parts, which would make the cooling process very long. A mathematical definition for a shape having maximum thickness no larger than d_{\max} is:

$$(9.10) \quad \forall x \in \Omega, \quad d_{\Omega}(x) \geq -\frac{d_{\max}}{2}.$$

This can be turned into an equality constraint, under the following form:

$$P_{\max T}(\Omega) = 0, \quad \text{where } P_{\max T}(\Omega) := \int_{\Omega} \left[d_{\Omega}(x) + \frac{d_{\max}}{2} \right]_-^2 dx,$$

where $[t]_- = \min(t, 0)$ is the negative part of a real number t .

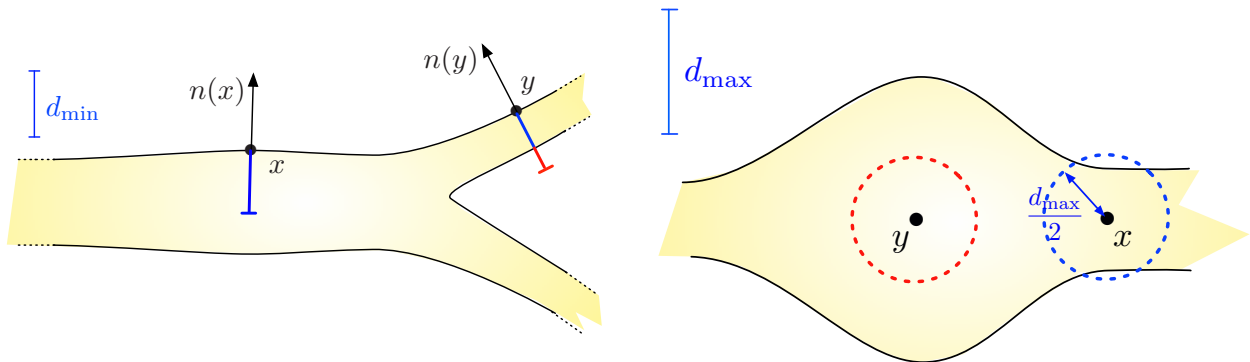


FIGURE 38. (Left) Example of one point x satisfying the minimum thickness constraint (9.9) and one point y violating it; (right) the point x satisfies the maximum thickness constraint (9.10) while the point y violates it.

Remark 9.1.

- The above formulations of thickness constraints are not the most efficient ones in numerical practice, but they conveniently allow to illustrate all the ideas in this section; see [134] for further details.
- The shape derivative of geometric constraints in (9.8) has a quite involved expression, especially because of the presence of the principal curvatures κ_i of $\partial\Omega$ and of integrals along rays. A variational method is introduced in the article [98] which allows to bypass the calculation of these quantities in the numerical evaluation of (9.8).

9.2.3. Numerical examples

We now provide two numerical examples, again in the context of structural mechanics (see Section 2.1.1), illustrating the effect of the maximum and minimum thickness constraints discussed in this section. These examples are excerpted from [134].

Our first example is concerned with maximum thickness control. We consider the 2d MBB Beam test case, as depicted in Figure 40 (top): the optimized shape belongs to a box D with size 3×1 . Its horizontal displacement vanishes on its left-hand side by symmetry, while its vertical displacement must vanish at its lower, right corner. A vertical load $g = (0, -0.5)$ is imposed on the upper left corner, and the compliance (2.7) of the structure is minimized, under a constraint on its volume, and on its maximum thickness. Several optimized shapes, associated to various values of the imposed maximum thickness d_{\max} , are represented in Figure 39.

Our next example, illustrating minimum thickness constraints, deals with compliant mechanisms. This is particularly interesting since such devices usually show very small hinges (mimicking ideal pointwise joints); we then expect difficulties in imposing a minimum length scale to such structures. The setting is that of a force inverter, as depicted in Figure 40 (top, left). The considered shapes are contained inside a 1×1 two-dimensional box D ; they are fixed at their upper and lower left corners, while a unit horizontal load $g = (1, 0)$ is applied at the center of their left-hand side. Our aim is that the horizontal displacement of the structure at the center x_T of their right-hand side be as negative as possible, so that the orientation of the imposed load g results in a displacement $u_\Omega(x_T)$ with opposite orientation. Several optimized shapes are depicted in Figure 40, associated to different values of the imposed minimum thickness d_{\min} . Obviously, the resulting device is less and less efficient as the value of d_{\min} is increased.

10. A GLIMPSE TO OTHER POPULAR SHAPE AND TOPOLOGY OPTIMIZATION FRAMEWORKS

Combining Hadamard's boundary variation method for shape sensitivity analysis with the level set method for the numerical representation of shapes and their deformations is only one among many shape and topology optimization frameworks, and in this section, we briefly sketch a few others.

In Section 10.1, we present the homogenization method. Although it is quite difficult to handle in general, it inspired the popular class of more heuristic density-based methods, which are discussed in Section 10.2. In the next Section 10.3, we sketch the phase field paradigm for shape and topology optimization, before turning in Section 10.4 to an original algorithm which solely relies on topological derivatives.

10.1. The homogenization method

As we have hinted at in Section 3, the homogenization method is based on the general trend of shape and topology optimization problems to favor thinner and thinner structures, with increasingly many holes, to the point that an optimal design may only be expected under the form of a composite material. More precisely, this method features a *relaxation* of the considered optimal design problem, where the set of admissible designs is enlarged so that it include such composite structures. The purpose of this section is to provide a formal intuition about this idea, referring to [8] and the monograph [4] for more rigorous and exhaustive introductions.

10.1.1. The model two-phase shape optimization problem

In order to ease the presentation, we slip into the context of the two-phase conductivity equation of Section 4.7. Let us recall from Section 7.3 that letting the conductivity of one of the phases tend to 0 allows (at least formally) to retrieve the one-phase-and-void shape optimization setting considered in most of this chapter.

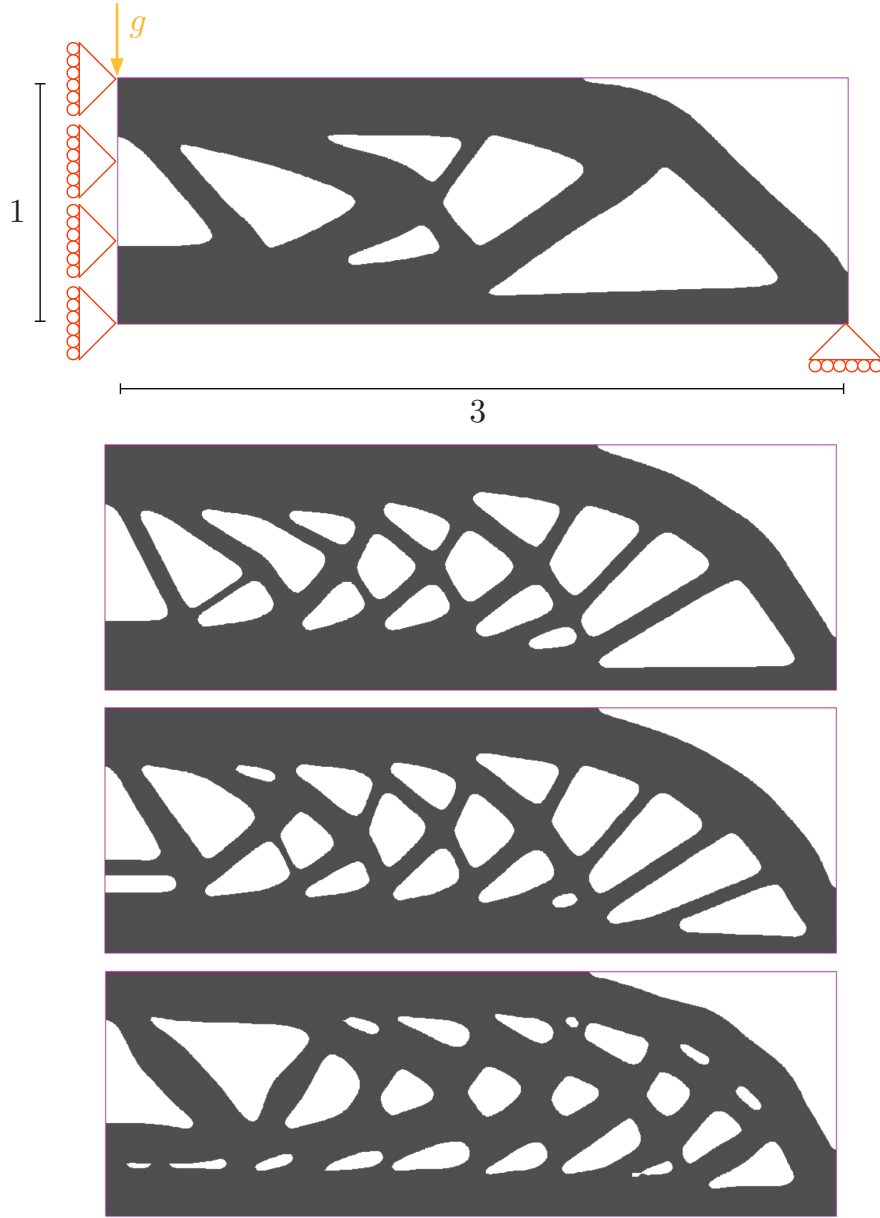


FIGURE 39. (From left to right, top to bottom) Optimized shape of a 2d MBB beam with no imposed maximum thickness, then $d_{\max} = 0.30, 0.25, 0.20$.

Introducing a fixed hold-all domain $D \subset \mathbb{R}^d$, we consider for each subset $\Omega \subset D$ the unique solution $u_\Omega \in H_0^1(D)$ to the equation:

$$(10.1) \quad \begin{cases} -\operatorname{div}(\gamma_\Omega \nabla u_\Omega) = f & \text{in } D, \\ u_\Omega = 0 & \text{on } \partial D, \end{cases} \quad \text{where } \gamma_\Omega(x) = \begin{cases} \beta & \text{if } x \in \Omega, \\ \alpha & \text{if } x \in D \setminus \Omega, \end{cases}$$

with $0 < \alpha < \beta$, and the source f is in $L^2(D)$. In this context, the shape optimization problem of interest reads:

$$(10.2) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega) \text{ s.t. } \operatorname{Vol}(\Omega) = V_T, \quad \text{where } J(\Omega) = \int_D j(u_\Omega) \, dx.$$

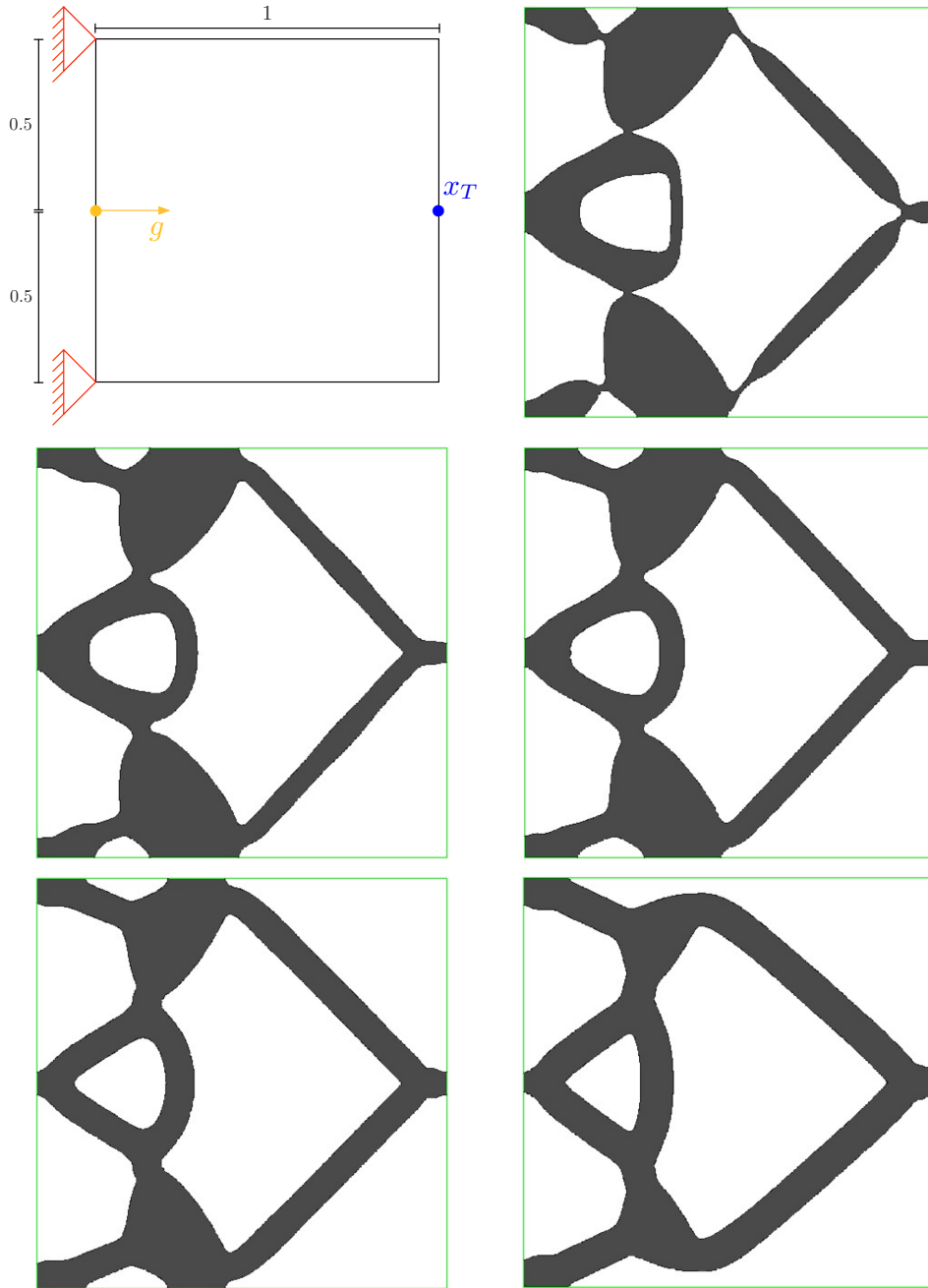


FIGURE 40. (Top left) Setting of the force inverter test case of Section 9.2.3; (from top to bottom, left to right) optimized shape with imposed minimum thickness $d_{\min} = 0, 0.05, 0.06, 0.07, 0.08$.

Here, $j : \mathbb{R} \rightarrow \mathbb{R}$ is a smooth function, satisfying the growth conditions (4.12), and V_T is a volume target. The admissible designs $\Omega \in \mathcal{U}_{\text{ad}}$ are measurable subsets of D , accounting for the repartition of the “pure phases” α and β within D . If $\Omega \subset \mathcal{U}_{\text{ad}}$ is identified with its characteristic function $\chi_\Omega \in L^\infty(D, \{0, 1\})$ given by:

$$\forall x \in D, \chi_\Omega(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

the set \mathcal{U}_{ad} can be equivalently rewritten:

$$\mathcal{U}_{\text{ad}} = \left\{ (\chi, \gamma) \in L^\infty(D, \{0, 1\}) \times L^\infty(D, \{\alpha, \beta\}) \text{ s.t. } \gamma(x) = \begin{cases} \alpha & \text{if } \chi(x) = 0 \\ \beta & \text{if } \chi(x) = 1 \end{cases} \text{ for a.e. } x \in D \right\}.$$

The homogenization method consists in extending \mathcal{U}_{ad} into the set $\mathcal{U}_{\text{ad}}^*$ of all *composite materials* obtained by mixing α and β :

$$(10.3) \quad \mathcal{U}_{\text{ad}}^* := \{(\rho, A) \in L^\infty(D, [0, 1]) \times L^\infty(D, \mathcal{M}_{\alpha, \beta}), \text{ s.t. } A(x) \in G_{\rho(x)}, x \in D\},$$

where we have denoted:

$$\mathcal{M}_{\alpha, \beta} := \{M \in \mathbb{R}_s^{d \times d}, \text{ s.t. } \forall \xi \in \mathbb{R}^d, \alpha |\xi|^2 \leq M \xi \cdot \xi \leq \beta |\xi|^2\},$$

and we recall that $\mathbb{R}_s^{d \times d}$ stands for the set of symmetric $d \times d$ matrices. In (10.3), $\mathcal{U}_{\text{ad}}^*$ gathers couples (ρ, A) composed of:

- A *density function* $\rho : D \rightarrow [0, 1]$ indicating the local volume fraction of the phase β at points $x \in D$.
- A *microstructure tensor* $A : D \rightarrow \mathcal{M}_{\alpha, \beta}$ which describes the local, microscopic arrangement of the two phases.

The microstructure tensor is subjected to the *material law* whereby, at any point $x \in D$, $A(x)$ has to belong to the strict subset $G_{\rho(x)} \subset \mathcal{M}_{\alpha, \beta}$, whose delicate mathematical definition is outlined in the next section; see [4] for details.

10.1.2. The set G_ρ of microstructure tensors

For a given value $\rho \in [0, 1]$, $G_\rho \subset \mathcal{M}_{\alpha, \beta}$ is the set of the *effective* (or averaged) conductivity tensors of all microscopic arrangements of the materials β and α in proportions ρ and $1 - \rho$, respectively: a matrix $A \in G_\rho$ is the limit (in some adapted sense, called *H-convergence* [141], [4]), and thereby the representative at the macroscopic level, of a conductivity coefficient which varies very rapidly at the microscopic scale.

Quite remarkably, the study of the effective conductivity tensors $A \in G_\rho$ can be achieved by considering only *periodic* arrangements of the phases α and β . Let $Y = (0, 1)^d$ be the unit periodicity cell; for any measurable subset $\omega \subset Y$ such that $|\omega| = \rho$,

$$(10.4) \quad A_\omega(y) = \left(\beta \chi_\omega(y) + \alpha (1 - \chi_\omega(y)) \right) \quad \text{for } y \in Y,$$

is the conductivity inside Y when ω is filled by the phase β and $Y \setminus \bar{\omega}$ is filled by α . In (10.4), it is understood that the characteristic function $\chi_\omega : Y \rightarrow \{0, 1\}$ and the matrix $A_\omega(y)$ are Y -periodic. The periodic repartition of α and β inside \mathbb{R}^d when the pattern (10.4) is reproduced at size $\varepsilon \ll 1$ is encoded by the conductivity coefficient:

$$A_\omega^\varepsilon(x) := A_\omega\left(\frac{x}{\varepsilon}\right), \quad x \in \mathbb{R}^d.$$

The effective conductivity matrix A_ω^* associated to this composite material is by definition the limit as $\varepsilon \rightarrow 0$ of the sequence A_ω^ε in the sense of *H-convergence*; see again [141] or [4]. The homogenization theory provides a closed form formula for A_ω^* :

$$(10.5) \quad (A_\omega^*)_{ij} = \int_Y A_\omega(y) (\nabla w_i + e_i) \cdot (\nabla w_j + e_j) dy, \quad i, j = 1, \dots, d,$$

where $e_i \in \mathbb{R}^d$ is the i^{th} basis vector and $w_i \in H^1(Y)$ is the unique solution (up to additive constants) to the so-called *cell problem*:

$$\begin{cases} -\text{div}(A_\omega(y)(\nabla w_i + e_i)) = 0 & \text{in } Y, \\ y \mapsto w_i(y) & \text{is } Y\text{-periodic.} \end{cases}$$

Finally, the set $G_\rho \subset \mathcal{M}_{\alpha, \beta}$ of the effective conductivity matrices describing microscopic arrangements of β and α in proportions ρ and $1 - \rho$ is defined as the closure in $\mathbb{R}_s^{d \times d}$ of the set of periodically homogenized matrices A_ω^* , given by formula (10.5), when $\omega \subset Y$ is measurable with $|\omega| = \rho$. Such a set has a very complicated structure. However, it turns out that G_ρ can be explicitly characterized as a subset of symmetric matrices in $\mathcal{M}_{\alpha, \beta}$ whose eigenvalues satisfy certain ρ -dependent bounds, called *Hashin-Shtrikman bounds* (see Theorem 2.2.3.1 in [4]). This precious fact is unfortunately very specific to the present two-phase conductivity setting: such an explicit characterization is not available, for instance, in the context of linearized

elasticity. Only some bounds on tensors in G_ρ are available in the latter case and the research of a sharper characterization of G_ρ is an open task, referred to as the *G-closure problem*.

Summarizing the previous discussion, the set $\mathcal{U}_{\text{ad}}^*$ of admissible composite materials defined in (10.3) is made of couples (ρ, A) such that, at every point $x \in D$, $A(x)$ is one of the possible effective behaviors of a periodic medium around x featuring an array of cells with small size $\varepsilon \ll 1$ where the two phases are arranged in proportions $\rho(x)$ and $(1 - \rho(x))$; see Figure 41 for an illustration. In particular, even if the original phases α and β are isotropic, due to their local arrangement in the cell, the resulting composite material A can be anisotropic, a feature which is crucial for optimality in many instances.

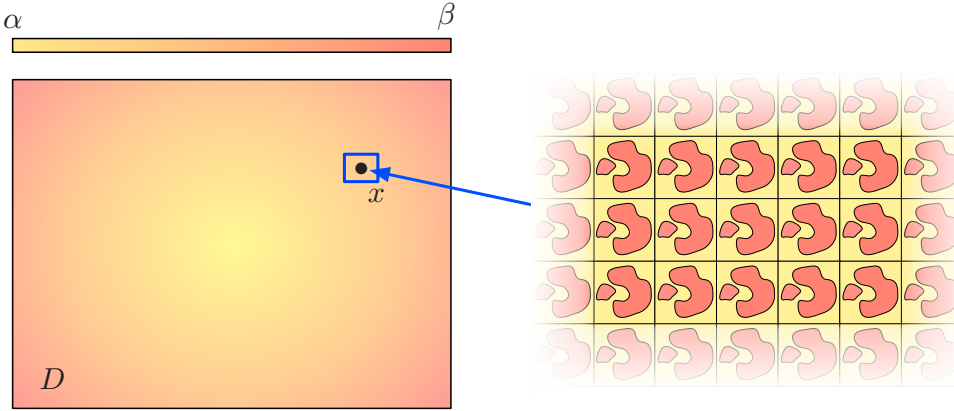


FIGURE 41. At the microscopic level around every point $x \in D$, a composite material resembles a periodic arrangement of the pure phases α and β with period $\varepsilon \ll 1$.

10.1.3. The relaxed, homogenized formulation of the original shape optimization problem

We are now in position to write down the relaxed version of the shape optimization problem (10.2):

$$(10.6) \quad \min_{(\rho, A) \in \mathcal{U}_{\text{ad}}^*} J^*(\rho, A) \text{ s.t. } \int_D \rho(x) \, dx = V_T, \text{ where } J^*(\rho, A) := \int_D j(u_{\rho, A}) \, dx,$$

and $u_{\rho, A}$ is the unique solution in $H_0^1(D)$ to the conductivity equation:

$$\begin{cases} -\operatorname{div}(A \nabla u_{\rho, A}) = f & \text{in } D, \\ u_{\rho, A} = 0 & \text{on } \partial D. \end{cases}$$

Many features of the new formulation (10.6) attest to the fact that it is the “good” way of extending the initial optimal design problem (10.2). Indeed, the following properties hold true:

- The problem (10.6) has at least one global solution $(\rho^*, A^*) \subset \mathcal{U}_{\text{ad}}^*$; intuitively, the set \mathcal{U}_{ad} in (10.2) has been sufficiently enriched so that the resulting optimal design problem has a solution.
- Every such solution is obtained as the limit (in the sense of H-convergence) of a sequence of admissible shapes $\Omega \in \mathcal{U}_{\text{ad}}$. The set $\mathcal{U}_{\text{ad}}^*$ is therefore “close enough” to the original problem (10.2) since no “artificial” solutions have been created in the enrichment process.

Let us emphasize, however, that the relaxed problem (10.6) may still have several local, non global minimum points. The relaxed formulation (10.6) of the shape optimization problem (10.2) lends itself to the numerical resolution. Indeed, the derivatives of the functional $J^*(\rho, A)$ with respect to the variables ρ and A are easily computed owing to the same adjoint techniques as in Section 4, so that it is tempting to apply a standard steepest descent algorithm. There is unfortunately a new difficulty in doing so which stems from the constraint whereby $A(x)$ should belong to the set $G_{\rho(x)}$ at any point $x \in D$. As we have seen indeed, G_ρ generally does not enjoy a nice, explicit characterization in terms of ρ . To overcome this problem, one possibility is to restrain the set G_ρ to a subset which is more easily parametrized in terms of ρ , thus limiting the type of the considered microstructures in (10.6). An interesting candidate for this operation is the subset

$L_\rho \subset G_\rho$ of the so-called *sequentially laminated composites*, which not only is easy to describe in terms of ρ , but also proves to be optimal in certain circumstances [4]. Another possibility is to choose a family of parametrized microstructures, like a rectangular hole in a square cell, possibly rotated, as was proposed in the seminal paper [43].

Let us conclude this presentation of the homogenization method with one limitation, which is actually common to all density-based methods. The optimized designs supplied by the relaxed formulation (10.6) are defined by a density field ρ and a microstructure tensor A , while in concrete situations, true “black-and-white” designs are desired, namely neat subsets of pure phases α and β . This raises the question of interpreting the produced design. One naive idea is simply to threshold the density ρ in such a way that the resulting shape have the desired volume V_T , but doing so is a little disappointing since it does not take advantage of how the optimized design looks like from a microscopic point of view (which certainly contains a lot of information). A better strategy is to penalize the intermediate densities by forcing ρ to be close to 0 or 1 (see the textbooks [4] and [45] for more details on this issue). Penalization is also a key issue for density-based topology optimization as discussed in the next section. Penalization works nicely on simple problems (like compliance minimization) but can be very tricky or even unsuccessful for problems involving complicate objective functions (like stress norms) or many constraints. A different and more subtle strategy was pioneered in [150], then elaborated on in [16, 105, 111], under the name of the *deshomogenization method*. In a nutshell, the data of the optimal density and microstructure tensor are used to construct a minimizing sequence of “black-and-white” designs which feature many small details. These intricate designs are now manufacturable thanks to the recent progress in additive manufacturing (where they are called *lattice materials*).

10.2. Density-based formulations

This section is a short overview of the popular density-based topology optimization techniques in engineering. These were originally introduced in the articles [42, 187], but a more complete account can be found in the book [45] and the recent survey [168]. The most popular algorithm in this class is the Solid Isotropic Material with Penalization (SIMP) method .

To illustrate our purpose, we retain the physical setting of the two-phase conductivity equation (10.1), in which we solve the shape optimization problem (10.2). The generalization of these ideas to the framework of linearized elasticity is fairly straightforward.

10.2.1. The density-based topology optimization problem

We have seen in Section 10.1 that the major difficulty posed by the homogenized version (10.6) of (10.2) is that of parametrizing the microstructure tensors $A \in G_\rho$, in terms of the density $\rho \in [0, 1]$.

Density-based topology optimization formulations are formal, simplified variants, which overcome this issue by assuming that for a given value of the density $\rho \in [0, 1]$, only one microstructure tensor $A(\rho)$ is possible, and that the material law $\rho \mapsto A(\rho)$ is known. This allows for a drastic simplification of the homogenized problem (10.6):

$$(10.7) \quad \min_{\rho \in L^\infty(D, [0,1])} J(\rho) \text{ s.t. } \int_D \rho \, dx = V_T, \text{ where } J(\rho) := \int_D j(u_\rho) \, dx,$$

and u_ρ is the unique solution in $H_0^1(D)$ to the equation:

$$(10.8) \quad \begin{cases} -\operatorname{div}(A(\rho)\nabla u_\rho) = f & \text{in } D, \\ u_\rho = 0 & \text{on } \partial D. \end{cases}$$

The formulation (10.7), (10.8) is that of a quite standard optimal control problem, where the design variable ρ is sought within a subset of a vector space. Simplified versions of the mathematical and numerical methods described in the previous sections allow to efficiently deal with such problems.

The apparent simplicity of this procedure hides several difficulties. At first, how should the material law $\rho \mapsto A(\rho)$ be defined? In close connection with this issue, how to guarantee that the optimized design does not contain “too many” intermediate densities?

10.2.2. The material interpolation law

In the seminal work [42], $A(\rho)$ was defined as the homogenized matrix A_ω^* in (10.5) induced by a square-shaped hole $\omega \in Y$ with measure $|\omega| = \rho$. This matrix was calculated using the finite element method for a few sample values ρ_1, \dots, ρ_N , then a continuous law $\rho \mapsto A(\rho)$ was inferred by interpolation. Soon, it became clear that, if intermediate densities were to be penalized, the precise value of the homogenized matrix $A(\rho)$ was not very relevant, except for optimization convergence purposes. Therefore, in practice, the definition of the material law $\rho \mapsto A(\rho)$ is rather guided by heuristic considerations. For instance, the law

$$(10.9) \quad A(\rho) = (\alpha + \rho^3(\beta - \alpha))\mathbf{I},$$

has soon been widely adopted under the name of Solid Isotropic Material with Penalization (SIMP) approach. Intuitively, it is expected that the presence of the power law in (10.9), together with a volume constraint, helps in driving the intermediate densities $\rho \in (0, 1)$ towards “black-and-white” values 0 or 1 in the course of the optimization process. Of course, the microstructure tensor $A(\rho)$ associated to intermediate densities $\rho \in (0, 1)$ may not exist in nature (one refers to a fictitious material approach) when such user-defined material laws are used, since it may not belong to G_ρ ; see [44] for a discussion about this point.

In many implementations, more complicated material laws than (10.9) are used, sharing the general structure:

$$(10.10) \quad A(\rho) = \zeta(L\rho)\mathbf{I}.$$

Here, $\zeta : \mathbb{R} \rightarrow \mathbb{R}$ is an *interpolation profile*, which assigns a conductivity coefficient to intermediate densities; it is a smooth function fulfilling the natural requirements:

$$\zeta(0) = \alpha, \text{ and } \zeta(1) = \beta.$$

The operator L is often referred to as a *filter* in the topology optimization literature. It may act on the optimized density function ρ in different manners depending on purpose. Here are a few examples (which can of course be combined):

- *Smoothing filters*: It is sometimes desirable to handle a regularized version $L\rho$ of the density field ρ . As proposed in [47], L can be a convolution operator which has a definite smoothing character and possibly a characteristic length scale $\ell > 0$ if the kernel in the convolution has, say, a compact support of size ℓ . Another possibility is to define $L\rho$ as the solution $q \in H^1(D)$ to the following equation:

$$\begin{cases} -\ell^2 \Delta q + q = h & \text{in } D, \\ \frac{\partial q}{\partial n} = 0 & \text{on } \partial D, \end{cases}$$

for a small value of the parameter $\ell > 0$, of the order of the mesh size in practice. From the early days of density-based topology optimization methods, smoothing the density field, and thereby imposing that it does not show too steep variations, has been identified as a means to avoid many numerical instabilities such as *checkerboards*; see [165]. Note that smoothing filters can also be defined at the discrete level, using the mesh discretization, for example by discrete convolution.

- *Heaviside filters*: As proposed in [112], filters can be devised with the purpose to drive the density field ρ towards “black-and-white” values 0 or 1. To achieve this, one possibility is to replace ρ by $H_\nu \rho$, where

$$H_\nu \rho = 1 - e^{-\nu \rho} + e^{-\nu} \rho.$$

Intuitively, the function H_ν gets closer and closer to the usual Heaviside function as $\nu \rightarrow +\infty$, so that during the optimization process, the intermediate values of ρ are attracted to either 0 or 1. Alternative version of such Heaviside filters, proving more efficient in practice, are described in [181].

- *Morphological filters*: When the original shape optimization problem features an objective (or constraint) function involving the geometry of the shape, it is desirable to formulate counterpart quantities in the context where the optimized design is a density field. For instance, *dilated* or *eroded* versions of a density field ρ have been proposed in [167], which come in handy in the formulation of geometric constraints, or when accounting for geometric uncertainties.

Finally, density-based techniques overcome the difficulty posed by the homogenization method by assuming a user-defined, fictitious material law. The difficulty of interpreting the optimized density field ρ as a “black-and-white” design remains, however. This is certainly an issue when it comes to constructing this design, but also in defining density-based counterparts to objective and constraint functions depending on the domain via geometric entities (see for instance Section 9.2) or involving complex physical phenomena such as fluid-structure interaction.

10.3. Phase field methods

Phase field, or “diffuse-interface” methods stem from physics, where they were tailored for the description of phase transition phenomena and complex interface dynamics, such as solidification or fracture propagation; see [137] for a historical overview and a general presentation of these methods. The phase field paradigm was introduced in the field of optimal design in [48]; see also the contributions [46, 49, 55, 80, 102, 175, 184]. There are also some variants of the phase field method, like the use of a reaction-diffusion equation [67], which are very similar. In this section, we illustrate its main features by dealing with the shape optimization problem (10.2) in the physical setting of the two-phase conductivity equation (10.1).

Phase-field methods are intrinsically meant to treat a version of the shape optimization problem (10.2) featuring an additional penalization of the perimeter of the interface $\partial\Omega$ between both phases:

$$(10.11) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} \left(J(\Omega) + \ell \text{Per}(\Omega) \right) \text{ s.t. } \text{Vol}(\Omega) = V_T,$$

where $\ell > 0$ is a fixed weighting factor. As we have hinted at in Section 3.2, this is a means to ensure that is has a global solution.

Like density-based methods, phase field methods reformulate the problem (10.11) as a material repartition problem, which is accounted for by a *phase field function* (referred to as “order parameter” in the physics literature) $\varphi : D \rightarrow \mathbb{R}$ such that:

$$\forall x \in D, \quad \begin{cases} \varphi(x) = 1 & \text{if there is only material } \beta \text{ around } x, \\ \varphi(x) = 0 & \text{if there is only material } \alpha \text{ around } x, \\ \varphi(x) \in (0, 1) & \text{if there is a mixture of both materials around } x. \end{cases}$$

A material law is chosen for assigning a conductivity value to intermediate densities, for instance:

$$A(\varphi) = \left(\alpha + \varphi(\beta - \alpha) \right) \mathbf{I},$$

so that the state equation (10.1) rewrites:

$$\begin{cases} -\text{div}(A(\varphi)\nabla u_\varphi) = f & \text{in } D, \\ u_\varphi = 0 & \text{on } \partial D. \end{cases}$$

The difference between phase field and density based methods lies in the mechanism whereby the design variable, φ or ρ respectively, is driven towards “black-and-white” values 0 or 1. As we have seen in Section 10.2, density-based methods usually rely on “clever” material laws to achieve this (including the use of filters). On the other hand, phase field methods feature a penalization of intermediate values, as well as of the length of the interface between both phases with the help of the so-called Van der Waals free energy functional:

$$F_\varepsilon(\varphi) = \int_D \left(\varepsilon^2 |\nabla \varphi|^2 + \frac{1}{\varepsilon^2} W(\varphi) \right) dx,$$

where ε is a small parameter, and $W : \mathbb{R} \rightarrow \mathbb{R}^+$ is a double-well potential with exactly two local minima at 0 and 1 such that $W(0) = W(1) = 0$, while $W(\varphi) > 0$ for $\varphi \neq 0, 1$. This type of functional is known to be a valuable approximation of the perimeter functional in the sense of Γ -convergence [3, 136]. Intuitively, as $\varepsilon \rightarrow 0$, the minimizers φ of F_ε are enticed to take values 0 or 1 so that the potential term stays bounded, while the penalization on the magnitude of $\nabla \phi$ become less and less stringent, so that ϕ can develop jumps.

All things considered the shape optimization problem (10.11) rewrites, using the phase field method:

$$(10.12) \quad \min_{\varphi \in L^\infty(D, [0,1])} J(\varphi) + \ell F_\varepsilon(\varphi) \text{ s.t. } \int_D \varphi dx = V_T, \text{ where } J(\varphi) = \int_D j(u_\varphi) dx.$$

At the theoretical level, one appealing feature of the above reformulation is that it is quite simple, owing to fairly standards methods in calculus of variations, to prove that for a fixed value ε , the above problem has a

global minimum; this is essentially due to the presence of the approximate perimeter term $F_\varepsilon(\varphi)$ in (10.12). It is also possible to prove that, under reasonable assumptions, global minimizers of the approximate problem (10.12) converge to global minimizers of the exact problem (10.11), regardless of the material interpolation rule; see [48]. Hence, provided ε is chosen to be sufficiently small, any such mixing rule can be used, in principle.

Remark 10.1. *Variants of this setting exist where the Van der Waals functional $F_\varepsilon(\varphi)$ is not added to the optimized functional $J(\varphi)$, but is used as the only minimized objective function. the definition of the double-well potential W is adapted so that the values of its two local minima depend on $J'(\varphi)$; see for instance [102, 175].*

From the practical point of view, the problem (10.12) falls into the context of optimal control, and it can be solved owing to simplified variants of the techniques presented in this chapter. Without entering too much into details about the resolution of (10.12), and omitting the volume constraint for simplicity, an optimal design is found as the limit of the sequence φ^n , $n \geq 0$ defined by:

$$(10.13) \quad \varphi^{n+1} = \varphi^n + \tau^n \widehat{\varphi}^n,$$

where τ^n is a descent step, and a descent direction $\widehat{\varphi}$ for (10.12) is inferred from the derivative of the objective function at φ via a change of inner products, along the lines of Section 5.2:

$$\text{Search for } \widehat{\varphi} \in H, \text{ s.t. } \forall \psi \in H, (\widehat{\varphi}, \psi)_H = J'(\varphi)(\psi) + \ell F'_\varepsilon(\varphi)(\psi).$$

Two popular choices in this regard are:

- $H = L^2(D)$, equipped with the standard inner product. The resulting gradient flow (i.e. the time continuous version of the update rule (10.13) is often referred to as the *Allen-Cahn equation*.
- $H = H^{-1}(D)$, equipped with the inner product:

$$(f_1, f_2)_{H^{-1}(D)} = \int_D \nabla u_1 \cdot \nabla u_2 \, dx, \text{ where } \begin{cases} -\Delta u_i = f_i & \text{in } D, \\ u_i = 0 & \text{on } \partial D. \end{cases}$$

The associated gradient flow is the *Cahn-Hilliard equation*.

Let us finally highlight that, like the homogenization method and density-based methods, the use of the phase field method for topology optimization imposes to reformulate the objective and constraint functionals featuring in the original shape optimization problem in this new context where the geometry of shapes is hidden, which may prove awkward in some applications.

10.4. An optimality criteria algorithm based on topological derivatives

We finally present another topology optimization algorithm which solely relies on topological derivatives. This algorithm was introduced in the article [29], and a more thorough mathematical analysis was conducted in [28].

Retaining the setting of the two-phase conductivity equation (10.1), we consider the shape and topology optimization problem (10.2), in a conveniently simplified version where the volume constraint is taken into account by means of a fixed penalization of the objective function:

$$(10.14) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} J(\Omega), \text{ where } J(\Omega) = \int_D j(u_\Omega) \, dx + \ell \text{Vol}(\Omega),$$

and $\ell > 0$ is a fixed weighting factor. The starting point of this method is the calculation of the topological derivative of the minimized function $J(\Omega)$. The Definition 7.1 of this notion, provided in Section 7.5, demands a little adaptation to the present two-phase setting. For any point $x \in D$, and $r > 0$ we let the variation $\Omega_{x,r}$ of Ω be:

$$\Omega_{x,r} = \begin{cases} \Omega \setminus \overline{B(x,r)} & \text{if } x \in \Omega, \\ \Omega \cup B(x,r) & \text{if } x \in D \setminus \overline{\Omega}; \end{cases}$$

that is, $\Omega_{x,r}$ is still obtained from Ω by nucleation of a tiny hole around x when $x \in \Omega$, but we also let the possibility to add a small bubble $B(x,r)$ to Ω when $x \notin \Omega$. The topological derivative $D_T J(\Omega)(x)$ of $J(\Omega)$ is now defined on the whole domain D , via the following asymptotic expansion, for $r > 0$ sufficiently small:

$$J(\Omega_{x,r}) = J(\Omega) + s_\Omega(x) r^d D_T J(\Omega)(x) + o(r^d), \text{ where } s_\Omega(x) := \begin{cases} 1 & \text{if } x \in \Omega, \\ -1 & \text{otherwise.} \end{cases}$$

With this notion at hand, one sufficient condition for a domain Ω to be a local minimizer of $J(\Omega)$ with respect to topological variations is:

- for all points $x \in \Omega$, one has: $D_T J(\Omega)(x) > 0$.
- for all points $x \in D \setminus \overline{\Omega}$, one has: $D_T J(\Omega)(x) < 0$.

In other words, Ω is a local minimum for $J(\Omega)$ if and only if $-D_T J(\Omega)$ is one possible level set function for Ω , in the sense that (7.1) holds, namely

$$\forall x \in D, \quad \begin{cases} -D_T J(\Omega)(x) < 0 & \text{if } x \in \Omega, \\ -D_T J(\Omega)(x) = 0 & \text{if } x \in \partial\Omega, \\ -D_T J(\Omega)(x) > 0 & \text{if } x \in D \setminus \overline{\Omega}. \end{cases}$$

Taking advantage of this remark, a local minimum for (10.14) can be achieved by the research of a function $\phi : D \rightarrow \mathbb{R}$ such that $\|\phi\|_{L^2(D)} = 1$ and

$$(10.15) \quad \phi = -\frac{1}{\|D_T J(\Omega)\|_{L^2(D)}} D_T J(\Omega), \text{ where } \Omega := \{x \in D, \phi(x) < 0\}.$$

Note that it is imposed that ϕ has unit norm in $L^2(D)$, without any loss of generality. Indeed, if ϕ is a level set function for a domain Ω , then so is $c\phi$, for any real number $c > 0$.

A function ϕ satisfying (10.15) may be found by applying a fixed point algorithm, with relaxation, adapted to functions with unit norm, following [164]. A sequence of functions ϕ^n is produced such that:

$$\phi^{n+1} = \frac{1}{\sin(a^n)} \left(\sin((1 - \tau^n)a^n)\phi^n + \sin(\tau^n a^n)g^n \right),$$

where:

- $g^n = -\frac{1}{\|D_T J(\Omega^n)\|_{L^2(D)}} D_T J(\Omega^n)$ is the normalized topological derivative of J at $\Omega^n := \{x \in D, \phi^n(x) < 0\}$;
- $a^n = \arccos((\phi^n, g^n)_{L^2(D)})$ is the angle between ϕ^n and g^n ;
- τ^n plays the role of a descent step.

This procedure is repeated until a suitable termination criterion is met.

Remark 10.2.

- *Although this algorithm features a representation of the optimized shape by a level set function, it is completely different from the level set method for shape and topology optimization presented in Section 7. Indeed, the measures of sensitivity of the optimized function with respect to the domain, and the update of the design between iterations are totally unrelated.*
- *This algorithm has paved the way to recent variants, making the connection between density-based methods and shape and topological sensitivity analyses; see [31, 99] for further details.*

11. PERSPECTIVES FOR SHAPE AND TOPOLOGY OPTIMIZATION

We conclude this chapter with a selection of two challenging topics, among others, arising from industrial applications, in close connection with optimal design. In Section 11.1, we discuss how the recent advent of additive manufacturing techniques in industry offers new perspectives in optimal design and raises new concerns. Then, Section 11.2 is a very brief, and biased, introduction to robust shape optimization where the optimized design must be equally optimal under uncertainties in some of the problem parameters.

11.1. Shape and topology optimization in connection with additive manufacturing

In light of the limitations imposed by traditional manufacturing processes on the optimized design, the recent and impressive headway in the development of additive manufacturing (or 3d printing) technologies have aroused a tremendous enthusiasm within the industrial community.

The wording ‘‘additive manufacturing’’ actually stands for a whole set of construction techniques with the common feature that they start with a slicing procedure, during which the produced shape (often supplied under the form of an STL mesh file) is divided into a series of horizontal layers, as in Figure 42 (top). These technologies then differ by how each successive layer is constructed; for instance,

- Fused Filament Fabrication techniques act by extrusion and deposition of a molten filament of material. These methods are often used to process plastic (ABS); see Figure 42 (bottom, left).

- Electron Beam Melting techniques are often used to process metal. A metallic powder bed is prepared within the construction chamber, and the layer under construction is drawn on the uppermost region by an electron beam which selectively heats and melts the powder; see Figure 42 (bottom, right).

We refer to the book [107] for a more complete presentation of the stakes of additive manufacturing.

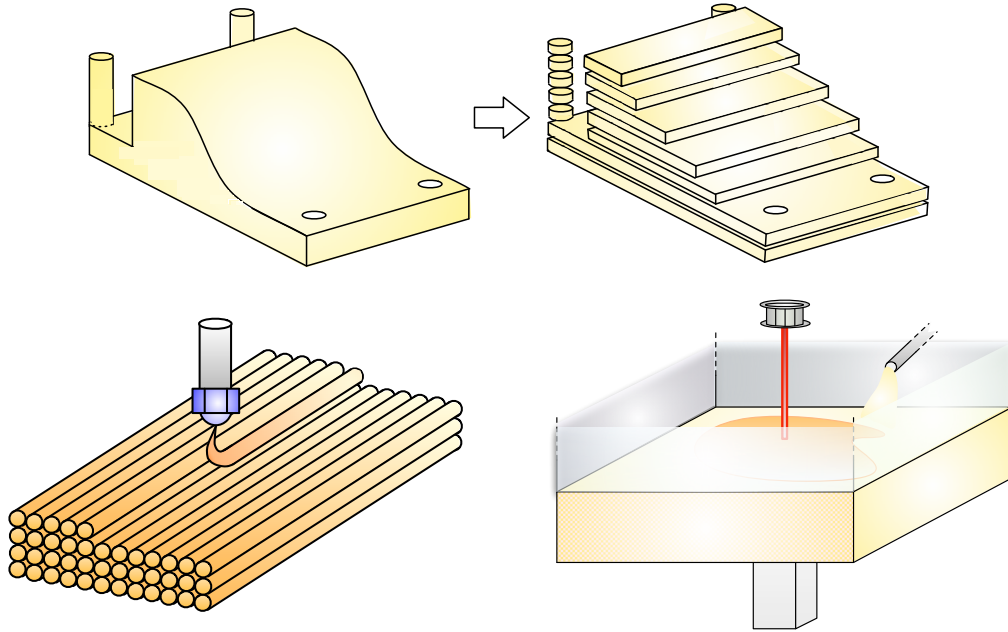


FIGURE 42. (Top) Illustration of the slicing procedure, initiating additive manufacturing process; (bottom, left) illustration of the Fused Filament Fabrication techniques; (bottom, right) illustration of the Electron Beam Melting techniques.

Additive manufacturing techniques are allegedly capable of assembling arbitrarily complicated structures, up to the resolution of the machine tool. In particular, they make it possible to assemble very thin lattice structures, such as the nearly optimal designs predicted by the homogenization theory (see Section 10.1 below). Nevertheless, they also suffer from drawbacks: beyond the needed assembly time, and the induced lack of scalability, they also impose constraints of their own on the assembled design, whose understanding and incorporation into the optimal design stage is key for unveiling their true potential in the industrial context. The most studied of them are outlined below, and we refer to the review article [131] for a more exhaustive presentation.

- All additive manufacturing technologies encounter difficulties when it comes to assemble shapes presenting large overhangs, that is, nearly horizontal regions hanging over void. Research in this direction has focused on either introducing constraints in the optimal design process so that the resulting shapes be self-supporting (see [12, 104, 125], among other works), or on the design of so-called *supports*, which are adapted *scaffold structures*, to be erected at the same time as the produced structure to sustain it during construction; see [7, 87] and references therein.
- Because all the constituent material of the shape is not melted and cooled at the same time, and because the layer by layer building process is fundamentally anisotropic, the material properties are different inside the layers or in the orthogonal built direction. As a consequence, the effective mechanical properties of structures built by additive manufacturing are inherently different from those produced by casting. It is a key issue to understand these effective properties (in connection with the production pattern), which are often anisotropic, and to take advantage of them for improving the performances of the final design.

- The difficulty to evacuate heat induced by the melting of the material may cause accumulation of thermal, residual stress inside the structure, causing the structure to warp during the cooling stage. Shape and topology optimization for minimizing these thermal effects during the building process, while maintaining good properties for the final use of the designed structure is very promising [17], although quite expensive because it is a typical multi-physics problem.
- Inner cavities are generally to be avoided, especially when powder bed construction methods are employed. Indeed, powder cannot be evacuated from such inner cavities.

11.2. Robust shape and topology optimization under uncertainties

The physical models underlying shape and topology optimization problems, involve physical parameters, whose precise values are often known imperfectly. For instance, in the context of a linearly elastic structure Ω as in Section 2.1.1:

- the body forces f , or the applied surface loads g are only accessed through error-prone measurements;
- the properties of the constituent material, i.e. its Lamé coefficients λ , μ , may be altered by changes in the conditions of the ambient medium (temperature, humidity, etc.);
- the geometry Ω of the structure itself may be subject to uncertainties, due to errors during the manufacturing process, or wear.

Usually, optimal design methodologies, such as those presented in this chapter, rely on the knowledge of the exact values of these parameters. Unfortunately, the performance of the optimized design may be strongly deteriorated in the case of perturbations of these parameters, however small. In the familiar context of the minimization of the compliance of elastic structures (see for instance Section 7.4), it even happens that the optimal design with respect to a given set of loads is the worst one when infinitesimally small perturbations of this load occur; see [66]. These observations call for the awareness of “small” uncertainties over physical parameters into the formulation of shape optimization problems.

To set ideas, let us consider the model unconstrained minimization problem (2.9) of a function $J(\Omega)$ of the domain. We denote by z the uncertain parameters of interest (for instance, z may stand for the loads f and g , or the Lamé parameters λ and μ , or even some features of the shape Ω itself), and by $J(\Omega, z)$ the value of the considered objective function when the parameters equal z . Depending on the context, two different paradigms can be thought of to incorporate a degree of robustness with respect to uncertainties over z into the shape optimization problem:

- When no information is available about the uncertain parameter z , but for a maximum amplitude m around an expected value z_0 , the *worst-case design* reformulation of (2.9) is in order, featuring the min-max problem:

$$(11.1) \quad \min_{\Omega \in \mathcal{U}_{\text{ad}}} \max_{\|z - z_0\| \leq m} J(\Omega, z).$$

In some very particular instances, this min-max problem can be conveniently transformed and tackled. This is the case when $J(\Omega)$ is the compliance of an elastic structure Ω and the uncertain parameter z stands for the forces f ; see [66, 79], and [30] for a related idea. In general, however, (11.1) has to be addressed as a bi-level optimization problem, which is very computationally expensive; see e.g. [113]. However, in the case of small uncertainties (namely small values of m), it is possible to greatly simplify (11.1) by linearizing the mapping $z \mapsto J(\Omega, z)$ around z_0 , and then the maximum in z is applied to this first order Taylor approximation of $J(\Omega, z)$, leading to an exact formula for this maximum, involving an adjoint state. Therefore (11.1) can be explicitly approximated by a simpler and tractable minimization problem; see [9, 114].

- When statistical information is available about the uncertain parameter z , a probabilistic description $z(\omega)$ may be used (where ω is the event variable). One may then optimize the mean value (or expectation with respect to ω) of the function $J(\Omega, z(\omega))$, as in

$$\min_{\Omega \in \mathcal{U}_{\text{ad}}} \mathbb{E}(J(\Omega, z(\omega))),$$

or higher-order moments of this quantity (such as its standard deviation). This type of problems can be tackled, e.g. by (costly) stochastic collocation methods [63, 132], or again in the case of small

uncertainties, by using a linearization of the mapping $z \mapsto J(\Omega, z)$ [127], or even a second-order Taylor expansion [10].

The question of how to cope with the presence of uncertainties in physical models in the perspective of optimal design is a very active research field; we refer to the survey [133] for an overview of the state-of-the-art around this issue.

Acknowledgements. This work is partly supported by the ANR Shapo project and by the SOFIA project, funded by Bpifrance (Banque Publique d’Investissement). G.A. is a member of the DEFI project at INRIA Saclay Ile-de-France. We are grateful to Marc Albertelli, Florian Feppon, Nicolas Lebbe and Georgios Michailidis for sharing some of their numerical results with us.

REFERENCES

- [1] R. ABGRALL, *Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes*, Communications on pure and applied mathematics, 49 (1996), pp. 1339–1373.
- [2] R. A. ADAMS AND J. J. FOURNIER, *Sobolev spaces*, vol. 140, Academic press, 2003.
- [3] G. ALBERTI, *Variational models for phase transitions, an approach via γ -convergence*, in Calculus of variations and partial differential equations, Springer, 2000, pp. 95–114.
- [4] G. ALLAIRE, *Shape optimization by the homogenization method*, vol. 146, Springer Science & Business Media, 2002.
- [5] G. ALLAIRE, *Conception optimale de structures*, vol. 58 of Mathématiques & Applications (Berlin) [Mathematics & Applications], Springer-Verlag, Berlin, 2007. With the collaboration of Marc Schoenauer (INRIA) in the writing of Chapter 8.
- [6] G. ALLAIRE, *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*, Oxford university press, 2007.
- [7] G. ALLAIRE AND B. BOGOSEL, *Optimizing supports for additive manufacturing*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 2493–2515.
- [8] G. ALLAIRE, E. BONNETIER, G. FRANCFORT, AND F. JOUVE, *Shape optimization by the homogenization method*, Numerische Mathematik, 76 (1997), pp. 27–68.
- [9] G. ALLAIRE AND C. DAPOGNY, *A linearized approach to worst-case design in parametric and geometric shape optimization*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 2199–2257.
- [10] ———, *A deterministic approximation method in shape optimization under random uncertainties*, SMAI Journal of computational mathematics, 1 (2015), pp. 83–143.
- [11] G. ALLAIRE, C. DAPOGNY, G. DELGADO, AND G. MICHAILIDIS, *Multi-phase structural optimization via a level set method*, ESAIM: Control, Optimisation and Calculus of Variations, 20 (2014), pp. 576–611.
- [12] G. ALLAIRE, C. DAPOGNY, R. ESTEVEZ, A. FAURE, AND G. MICHAILIDIS, *Structural optimization under overhang constraints imposed by additive manufacturing technologies*, Journal of Computational Physics, 351 (2017), pp. 295–328.
- [13] G. ALLAIRE, C. DAPOGNY, AND P. FREY, *Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh*, Comptes Rendus Mathématique, 349 (2011), pp. 999–1003.
- [14] ———, *Shape optimization with a level set based mesh evolution method*, Computer Methods in Applied Mechanics and Engineering, 282 (2014), pp. 22–53.
- [15] G. ALLAIRE, F. DE GOURNAY, F. JOUVE, AND A.-M. TOADER, *Structural optimization using topological and shape sensitivity via a level set method*, Control and Cybernetics, 34 (2005), p. 59.
- [16] G. ALLAIRE, P. GEOFFROY-DONDERS, AND O. PANTZ, *Topology optimization of modulated and oriented periodic microstructures by the homogenization method*, Computers & Mathematics with Applications, 78 (2019), pp. 2197–2229.
- [17] G. ALLAIRE AND L. JAKABČIN, *Taking into account thermal residual stresses in topology optimization of structures built by additive manufacturing*, M3AS, 28 (2018), pp. 2313–2366.
- [18] G. ALLAIRE AND F. JOUVE, *A level-set method for vibration and multiple loads structural optimization*, Computer methods in applied mechanics and engineering, 194 (2005), pp. 3269–3290.
- [19] ———, *Minimum stress optimal design with the level set method*, Engineering analysis with boundary elements, 32 (2008), pp. 909–918.
- [20] G. ALLAIRE, F. JOUVE, AND G. MICHAILIDIS, *Thickness control in structural optimization via a level set method*, Structural and Multidisciplinary Optimization, 53 (2016), pp. 1349–1382.
- [21] G. ALLAIRE, F. JOUVE, AND A.-M. TOADER, *A level-set method for shape optimization*, Comptes Rendus Mathématique, 334 (2002), pp. 1125–1130.
- [22] ———, *Structural optimization using sensitivity analysis and a level-set method*, Journal of computational physics, 194 (2004), pp. 363–393.
- [23] G. ALLAIRE, F. JOUVE, AND N. VAN GOETHEM, *Damage and fracture evolution in brittle materials by shape optimization methods*, J. Comp. Phys., 230 (2011), pp. 5010–5044.
- [24] G. ALLAIRE, A. KARRMAN, AND G. MICHAILIDIS, *A 2-d level set code for shape and topology optimization, available at: <http://www.cmap.polytechnique.fr/~allaire/levelset.en.html>*, 2009.
- [25] G. ALLAIRE AND O. PANTZ, *Structural optimization with FreeFem++*, Structural and Multidisciplinary Optimization, 32 (2006), pp. 173–181.

- [26] L. AMBROSIO AND G. BUTTAZZO, *An optimal design problem with perimeter penalization*, Calculus of Variations and Partial Differential Equations, 1 (1993), pp. 55–69.
- [27] H. AMMARI, S. MOSKOW, AND M. S. VOGELIUS, *Boundary integral formulae for the reconstruction of electric and electromagnetic inhomogeneities of small volume*, ESAIM: Control, Optimisation and Calculus of Variations, 9 (2003), pp. 49–66.
- [28] S. AMSTUTZ, *Analysis of a level set method for topology optimization*, Optimization Methods and Software, 26 (2011), pp. 555–573.
- [29] S. AMSTUTZ AND H. ANDRÄ, *A new algorithm for topology optimization using a level-set method*, Journal of Computational Physics, 216 (2006), pp. 573–588.
- [30] S. AMSTUTZ AND M. CILIGOT-TRAVAIN, *A notion of compliance robustness in topology optimization*, ESAIM: Control, Optimisation and Calculus of Variations, 22 (2016), pp. 64–87.
- [31] S. AMSTUTZ, C. DAPOGNY, AND À. FERRER, *A consistent relaxation of optimal design problems for coupling shape and topological derivatives*, Numerische Mathematik, (2016), pp. 1–60.
- [32] M. ANDERSEN, J. DAHL, AND L. VANDENBERGHE, *Cvxopt: A python package for convex optimization*, cvxopt.org, (2013).
- [33] E. ANDREASSEN, A. CLAUSEN, M. SCHEVENELS, B. S. LAZAROV, AND O. SIGMUND, *Efficient topology optimization in matlab using 88 lines of code*, Structural and Multidisciplinary Optimization, 43 (2011), pp. 1–16.
- [34] S. ARGUILLÈRE, E. TRÉLAT, A. TROUVÉ, AND L. YOUNES, *Shape deformation analysis from the optimal control viewpoint*, J. Math. Pures Appl. (9), 104 (2015), pp. 139–178.
- [35] G. AUBERT, M. BARLAUD, O. FAUGERAS, AND S. JEHAN-BESSON, *Image segmentation using active contours: Calculus of variations or shape gradients?*, SIAM Journal on Applied Mathematics, 63 (2003), pp. 2128–2154.
- [36] J.-F. AUJOL AND G. AUBERT, *Signed distance functions and viscosity solutions of discontinuous Hamilton-Jacobi equations*, INRIA report RR-4507, (2002).
- [37] T. BAKER AND P. CAVALLO, *Dynamic adaptation for deforming tetrahedral meshes*, in 14th Computational Fluid Dynamics Conference, 1999, p. 3253.
- [38] C. BARBAROSIE, S. LOPES, AND A.-M. TOADER, *An algorithm for constrained optimization with applications to the design of mechanical structures*, in International Conference on Engineering Optimization, Springer, 2018, pp. 272–284.
- [39] T. J. BARTH AND J. A. SETHIAN, *Numerical schemes for the hamilton-jacobi and level set equations on triangulated domains*, Journal of Computational Physics, 145 (1998), pp. 1–40.
- [40] L. L. BEGHINI, A. BEGHINI, N. KATZ, W. F. BAKER, AND G. H. PAULINO, *Connecting architecture and engineering through structural topology optimization*, Engineering Structures, 59 (2014), pp. 716–726.
- [41] G. BELLETTINI, *Lecture notes on mean curvature flow, barriers and singular perturbations*, vol. 12 of Appunti. Scuola Normale Superiore di Pisa (Nuova Serie) [Lecture Notes. Scuola Normale Superiore di Pisa (New Series)], Edizioni della Normale, Pisa, 2013.
- [42] M. P. BENDSOE, *Optimal shape design as a material distribution problem*, Struct. Optim., 1 (1989), pp. 193–202.
- [43] M. P. BENDSOE AND N. KIKUCHI, *Generating optimal topologies in structural design using a homogenization method*, Computer methods in applied mechanics and engineering, 71 (1988), pp. 197–224.
- [44] M. P. BENDSOE AND O. SIGMUND, *Material interpolation schemes in topology optimization*, Archive of applied mechanics, 69 (1999), pp. 635–654.
- [45] ———, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.
- [46] L. BLANK, H. GARCKE, L. SARBU, T. SRISUPATTARAWANIT, V. STYLES, AND A. VOIGT, *Phase-field approaches to structural topology optimization*, in Constrained optimization and optimal control for partial differential equations, Springer, 2012, pp. 245–256.
- [47] B. BOURDIN, *Filters in topology optimization*, International journal for numerical methods in engineering, 50 (2001), pp. 2143–2158.
- [48] B. BOURDIN AND A. CHAMBOLLE, *Design-dependent loads in topology optimization*, ESAIM: Control, Optimisation and Calculus of Variations, 9 (2003), pp. 19–48.
- [49] ———, *The phase-field method in optimal design*, in IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, Springer, 2006, pp. 207–215.
- [50] H. BREZIS, *Functional analysis, Sobolev spaces and partial differential equations*, Springer Science & Business Media, 2010.
- [51] D. BUCUR AND G. BUTTAZZO, *Variational methods in shape optimization problems*, Progress in Nonlinear Differential Equations and their Applications, 65., Birkhäuser, 2005.
- [52] C. BUI, C. DAPOGNY, AND P. FREY, *An accurate anisotropic adaptation method for solving the level set advection equation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 899–922.
- [53] M. BURGER, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.
- [54] M. BURGER, B. HACKL, AND W. RING, *Incorporating topological derivatives into level set methods*, Journal of Computational Physics, 194 (2004), pp. 344–362.
- [55] M. BURGER AND R. STAINKO, *Phase-field relaxation of topology optimization with local stress constraints*, SIAM Journal on Control and Optimization, 45 (2006), pp. 1447–1466.
- [56] E. BURMAN, D. ELFVÉRSO, P. HANSBO, M. G. LARSON, AND K. LARSSON, *Cut topology optimization for linear elasticity with coupling to parametric nondesign domain regions*, Computer Methods in Applied Mechanics and Engineering, 350 (2019), pp. 462–479.

- [57] E. CANCÈS, R. KERIVEN, F. LODIER, AND A. SAVIN, *How electrons guard the space: shape optimization with probability distribution criteria*, Theoretical Chemistry Accounts, 111 (2004), pp. 373–380.
- [58] P. CANNARSA AND P. CARDALIAGUET, *Representation of equilibrium solutions to the table problem of growing sandpiles*, Journal of the European Mathematical Society, 6 (2004), pp. 435–464.
- [59] J. CÉA, *Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût*, ESAIM: Mathematical Modelling and Numerical Analysis, 20 (1986), pp. 371–402.
- [60] V. J. CHALLIS, *A discrete level-set topology optimization code written in Matlab*, Struct. Multidiscip. Optim., 41 (2010), pp. 453–464.
- [61] V. J. CHALLIS AND J. K. GUEST, *Level set topology optimization of fluids in stokes flow*, International journal for numerical methods in engineering, 79 (2009), pp. 1284–1308.
- [62] A. CHAMBOLLE, *A density result in two-dimensional linearized elasticity and applications*, Arch. Ration. Mech. Anal., 167 (2003), pp. 211–233.
- [63] S. CHEN, W. CHEN, AND S. LEE, *Level set based robust shape and topology optimization under random field uncertainties*, Structural and Multidisciplinary Optimization, 41 (2010), pp. 507–524.
- [64] D. CHENAIS, *On the existence of a solution in a domain identification problem*, J. Math. Anal. Appl., 52 (1975), pp. 189–289.
- [65] A. CHERKAEV, *Variational methods for structural optimization*, vol. 140, Springer Science & Business Media, 2012.
- [66] A. CHERKAEV AND E. CHERKAEVA, *Optimal design for uncertain loading condition*, in Homogenization: In Memory of Serguei Kozlov, World Scientific, 1999, pp. 193–213.
- [67] J. S. CHOI, T. YAMADA, K. IZUI, S. NISHIWAKI, AND J. YOO, *Topology optimization using a reaction-diffusion equation*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2407–2420.
- [68] D. L. CHOPP, *Computing minimal surfaces via level set curvature flow*, Journal of Computational Physics, 106 (1993), pp. 77–91.
- [69] A. N. CHRISTIANSEN, J. A. BÆRENTZEN, M. NOBEL-JØRGENSEN, N. AAGE, AND O. SIGMUND, *Combined shape and topology optimization of 3d structures*, Computers & Graphics, 46 (2015), pp. 25–35.
- [70] A. N. CHRISTIANSEN, M. NOBEL-JØRGENSEN, N. AAGE, O. SIGMUND, AND J. A. BÆRENTZEN, *Topology optimization using an explicit interface representation*, Structural and Multidisciplinary Optimization, 49 (2014), pp. 387–399.
- [71] P. G. CIARLET, *The finite element method for elliptic problems*, vol. 40, Siam, 2002.
- [72] C. DAPOGNY, C. DOBRZYNSKI, AND P. FREY, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, Journal of computational physics, 262 (2014), pp. 358–378.
- [73] C. DAPOGNY, C. DOBRZYNSKI, P. FREY, AND A. FROELHY, *mmg*, available at: <https://www.mmgtools.org>, 2019.
- [74] C. DAPOGNY AND P. FREY, *Computation of the signed distance function to a discrete contour on adapted triangulation*, Calcolo, 49 (2012), pp. 193–219.
- [75] C. DAPOGNY, P. FREY, AND A. FROELHY, *ISCD Toolbox*, <https://github.com/ISCDtoolbox>, 2019.
- [76] C. DAPOGNY, P. FREY, F. OMNÈS, AND Y. PRIVAT, *Geometrical shape optimization in fluid mechanics using freefem++*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 2761–2788.
- [77] C. DAPOGNY, N. LEBBE, AND E. OUDET, *Optimization of the shape of regions supporting boundary conditions*, Hal preprint: <https://hal.archives-ouvertes.fr/hal-02064477/>, (2019).
- [78] F. DE GOURNAY, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM journal on control and optimization, 45 (2006), pp. 343–367.
- [79] F. DE GOURNAY, G. ALLAIRE, AND F. JOUVE, *Shape and topology optimization of the robust compliance via the level set method*, ESAIM: Control, Optimisation and Calculus of Variations, 14 (2008), pp. 43–70.
- [80] L. DEDÈ, M. J. BORDEN, AND T. J. HUGHES, *Isogeometric analysis for topology optimization with a phase field model*, Archives of Computational Methods in Engineering, 19 (2012), pp. 427–465.
- [81] M. DELFOUR AND J.-P. ZOLÉSIO, *Shape identification via metrics constructed from the oriented distance function*, Control and Cybernetics, 34 (2005), pp. 137–164.
- [82] ———, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, SIAM, 2011.
- [83] C. DOBRZYNSKI, *Adaptation de maillage anisotrope 3d et application à l'aéro-thermique des bâtiments*, PhD thesis, Université Pierre et Marie Curie, Paris 6, 2005.
- [84] C. DOBRZYNSKI AND P. FREY, *Anisotropic delaunay mesh adaptation for unsteady simulations*, in Proceedings of the 17th international Meshing Roundtable, Springer, 2008, pp. 177–194.
- [85] G. DOĞAN, *Shape calculus for shape energies in image processing*, arXiv preprint arXiv:1307.5797, (2013).
- [86] X.-B. DUAN, Y.-C. MA, AND R. ZHANG, *Shape-topology optimization of stokes flow via variational level set method*, Applied Mathematics and Computation, 202 (2008), pp. 200–209.
- [87] J. DUMAS, J. HERGEL, AND S. LEFEBVRE, *Bridging the gap: automated steady scaffoldings for 3d printing*, ACM Transactions on Graphics (TOG), 33 (2014), pp. 1–10.
- [88] P. D. DUNNING AND A. H. KIM, *Introducing the sequential linear programming level-set method for topology optimization*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 631–643.
- [89] P. DUYSINX, L. VAN MIEGROET, T. JACOBS, AND C. FLEURY, *Generalized shape optimization using x-fem and level set methods*, in IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, Springer, 2006, pp. 23–32.
- [90] A. ERN AND J.-L. GUERMOND, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media, 2013.

- [91] H. ESCHENAUER, V. KOBELEV, AND A. SCHUMACHER, *Bubble method for topology and shape optimization of structures*, Structural Optimization, 8 (1994), pp. 42–51.
- [92] L. C. EVANS AND R. F. GARIEPY, *Measure theory and fine properties of functions*, CRC press, 2015.
- [93] S. FATHPOUR AND B. JALALI, *Silicon photonics for telecommunications and biomedicine*, CRC Press, 2011.
- [94] A. FAURE, *Optimisation de forme de matériaux et structures architecturés par la méthode des lignes de niveaux avec prise en compte des interfaces graduées*, PhD thesis, Grenoble Alpes, 2017.
- [95] F. FEPPON, *Shape and topology optimization of multiphysics systems*, PhD thesis, Thèse de doctorat de l'Université Paris Saclay préparée à l'École polytechnique, 2019.
- [96] F. FEPPON, G. ALLAIRE, F. BORDEU, J. CORTIAL, AND C. DAPOGNY, *Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework*, SeMA Journal, 76(3) (2019), pp. 413–458.
- [97] F. FEPPON, G. ALLAIRE, AND C. DAPOGNY, *Null space gradient flows for constrained optimization with applications to shape optimization*, submitted, Hal preprint: <https://hal.archives-ouvertes.fr/hal-01972915/>, (2019).
- [98] ———, *A variational formulation for computing shape derivatives of geometric constraints along rays*, ESAIM: M2AN, 54 (2020), pp. 181–228.
- [99] À. FERRER, *Simp-all: A generalized simp method based on the topological derivative concept*, International Journal for Numerical Methods in Engineering, 120 (2019), pp. 361–381.
- [100] G. FREMIOT AND J. SOKOLOWSKI, *Shape sensitivity analysis of problems with singularities*, Lecture notes in pure and applied mathematics, (2001), pp. 255–276.
- [101] P. FREY AND P.-L. GEORGE, *Mesh generation: application to finite elements*, ISTE, 2007.
- [102] A. L. GAIN AND G. H. PAULINO, *Phase-field based topology optimization with polygonal elements: a finite volume approach for the evolution equation*, Structural and Multidisciplinary Optimization, 46 (2012), pp. 327–342.
- [103] S. GARREAU, P. GUILLAUME, AND M. MASMOUDI, *The topological asymptotic for pde systems: the elasticity case*, SIAM journal on control and optimization, 39 (2001), pp. 1756–1778.
- [104] A. T. GAYNOR AND J. K. GUEST, *Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design*, Structural and Multidisciplinary Optimization, 54 (2016), pp. 1157–1172.
- [105] P. GEOFFROY-DONDERS, G. ALLAIRE, AND O. PANTZ, *3-d topology optimization of modulated and oriented periodic microstructures by the homogenization method*, Journal of Computational Physics, 401 (2020), p. 108994.
- [106] M. GIACOMINI, O. PANTZ, AND K. TRABELSI, *Volumetric expressions of the shape gradient of the compliance in structural shape optimization*, arXiv preprint arXiv:1701.05762, (2017).
- [107] I. GIBSON, D. ROSEN, AND B. STUCKER, *Additive manufacturing technologies*, vol. 17, Springer, 2015.
- [108] Y. GIGA, *Surface evolution equations*, Springer, 2006.
- [109] J. C. GILBERT, G. LE VEY, AND J. MASSE, *La différentiation automatique de fonctions représentées par des programmes*, INRIA report RR-1557, (1991).
- [110] P. GRISVARD, *Elliptic problems in nonsmooth domains*, SIAM, 2011.
- [111] J. P. GROEN AND O. SIGMUND, *Homogenization-based topology optimization for high-resolution manufacturable microstructures*, International Journal for Numerical Methods in Engineering, 113 (2018), pp. 1148–1163.
- [112] J. K. GUEST, J. H. PRÉVOST, AND T. BELYTSCHKO, *Achieving minimum length scale in topology optimization using nodal design variables and projection functions*, International journal for numerical methods in engineering, 61 (2004), pp. 238–254.
- [113] X. GUO, W. BAI, W. ZHANG, AND X. GAO, *Confidence structural robust design and optimization under stiffness and load uncertainties*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 3378–3399.
- [114] X. GUO, W. ZHANG, AND L. ZHANG, *Robust structural topology optimization considering boundary uncertainties*, Computer Methods in Applied Mechanics and Engineering, 253 (2013), pp. 356–368.
- [115] X. GUO, W. ZHANG, AND W. ZHONG, *Explicit feature control in structural topology optimization via level set method*, Computer Methods in Applied Mechanics and Engineering, 272 (2014), pp. 354–378.
- [116] M. E. GURTIN, *An introduction to continuum mechanics*, vol. 158, Academic press, 1982.
- [117] J. HADAMARD, *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, in Mémoires présentés par divers savants à l'Académie des Sciences, vol. 33, Imprimerie nationale, 1908.
- [118] A. HENROT AND M. PIERRE, *Shape Variation and Optimization*, EMS Tracts in Mathematics Vol. 28, 2018.
- [119] R. HIPTMAIR, A. PAGANINI, AND S. SARGHEINI, *Comparison of approximate shape gradients*, BIT Numerical Mathematics, 55 (2015), pp. 459–485.
- [120] J. JENSEN AND O. SIGMUND, *Topology optimization for nano-photonics*, Laser & Photonics Reviews, 5 (2001), pp. 308–321.
- [121] F. JOUVE AND H. MECHKOUR, *Level set based method for design of compliant mechanisms*, European Journal of Computational Mechanics, 17 (2008), pp. 957–968.
- [122] R. KIMMEL AND J. A. SETHIAN, *Computing geodesic paths on manifolds*, Proceedings of the national academy of Sciences, 95 (1998), pp. 8431–8435.
- [123] R. V. KOHN AND G. W. MILTON, *On bounding the effective conductivity of anisotropic composites*, in Homogenization and effective moduli of materials and media, Springer, 1986, pp. 97–125.
- [124] S. LANG, *Fundamentals of differential geometry*, vol. 191, Springer Science & Business Media, 2012.
- [125] M. LANGELAAR, *Topology optimization of 3d self-supporting structures for additive manufacturing*, Additive Manufacturing, 12 (2016), pp. 60–70.

- [126] A. LAURAIN, *A level set-based structural optimization code using fenics*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 1311–1334.
- [127] B. S. LAZAROV, M. SCHEVENELS, AND O. SIGMUND, *Topology optimization with geometric uncertainties by perturbation techniques*, International Journal for Numerical Methods in Engineering, 90 (2012), pp. 1321–1336.
- [128] N. LEBBE, C. DAPOGNY, E. OUDET, K. HASSAN, AND A. GLIERE, *Robust shape and topology optimization of nanophotonic devices using the level set method*, Journal of Computational Physics, 395 (2019), pp. 710–746.
- [129] R. J. LEVEQUE, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98, Siam, 2007.
- [130] J.-L. LIONS, *Optimal control of systems governed by partial differential equations*, Grundlehren der mathematischen Wissenschaften, Springer-Verlag, 1971.
- [131] J. LIU, A. T. GAYNOR, S. CHEN, Z. KANG, K. SURESH, A. TAKEZAWA, L. LI, J. KATO, J. TANG, C. C. WANG, ET AL., *Current and future trends in topology optimization for additive manufacturing*, Structural and Multidisciplinary Optimization, 57 (2018), pp. 2457–2483.
- [132] J. MARTÍNEZ-FRUTOS, M. KESSLER, AND F. PERIAGO, *Robust optimal shape design for an elliptic pde with uncertainty in its input data*, ESAIM: Control, Optimisation and Calculus of Variations, 21 (2015), pp. 901–923.
- [133] K. MAUTE, *Topology optimization under uncertainty*, in Topology optimization in structural and continuum mechanics, Springer, 2014, pp. 457–471.
- [134] G. MICHAÏLIDIS, *Manufacturing constraints and multi-phase shape and topology optimization via a level-set method*, PhD thesis, Ecole Polytechnique, 2014.
- [135] I. M. MITCHELL, *The flexible, extensible and efficient toolbox of level set methods*, Journal of Scientific Computing, 35 (2008), pp. 300–329.
- [136] L. MODICA AND S. MORTOLA, *Un esempio di γ -convergenza*, Unione Mat. Ital. Sez. B, 14 (1977), pp. 285–299.
- [137] N. MOELANS, B. BLANPAIN, AND P. WOLLANTS, *An introduction to phase-field modeling of microstructure evolution*, Calphad, 32 (2008), pp. 268–294.
- [138] B. MOHAMMADI AND O. PIRONNEAU, *Applied shape optimization for fluids*, Oxford university press, 2010.
- [139] F. MURAT AND J. SIMON, *Etudes de problèmes d’optimal design*, in Lecture Notes in Computer Science, vol. 41, Springer, 1976, pp. 54–62.
- [140] ———, *Sur le contrôle par un domaine géométrique*, Pré-publication du Laboratoire d’Analyse Numérique, 76015, (1976).
- [141] F. MURAT AND L. TARTAR, *H-convergence*, in Topics in the mathematical modelling of composite materials, vol. 31 of Progr. Nonlinear Differential Equations Appl., Birkhäuser Boston, Boston, MA, 1997, pp. 21–43.
- [142] E. J. NIELSEN AND W. K. ANDERSON, *Recent improvements in aerodynamic design optimization on unstructured meshes*, AIAA journal, 40 (2002), pp. 1155–1163.
- [143] J. NOCEDAL AND S. J. WRIGHT, *Numerical optimization 2nd*, Springer, 2006.
- [144] A. A. NOVOTNY AND J. SOKOŁOWSKI, *Topological derivatives in shape optimization*, Springer Science & Business Media, 2012.
- [145] S. OSHER AND R. FEDKIW, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Science & Business Media, 2006.
- [146] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.
- [147] S. OSHER AND C.-W. SHU, *High-order essentially nonoscillatory schemes for hamilton-jacobi equations*, SIAM Journal on numerical analysis, 28 (1991), pp. 907–922.
- [148] S. J. OSHER AND F. SANTOSA, *Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum*, Journal of Computational Physics, 171 (2001), pp. 272–288.
- [149] O. PANTZ, *Sensibilité de l’équation de la chaleur aux sauts de conductivité*, Comptes Rendus Mathématique, 341 (2005), pp. 333–337.
- [150] O. PANTZ AND K. TRABELSI, *A post-treatment of the homogenization method for shape optimization*, SIAM Journal on Control and Optimization, 47 (2008), pp. 1380–1398.
- [151] J. PARK, A. SUTRADHAR, J. J. SHAH, AND G. H. PAULINO, *Design of complex bone internal structure using topology optimization with perimeter control*, Computers in biology and medicine, 94 (2018), pp. 74–84.
- [152] O. PIRONNEAU, *Optimal shape design for elliptic systems*, Springer, 1982.
- [153] ———, *Finite element methods for fluids*, Wiley Chichester, 1989.
- [154] L. S. PONTRYAGIN, *Mathematical theory of optimal processes*, Routledge, 2018.
- [155] T. A. POULSEN, *A new scheme for imposing a minimum length scale in topology optimization*, International Journal for Numerical Methods in Engineering, 57 (2003), pp. 741–760.
- [156] B. PROTAS, T. R. BEWLEY, AND G. HAGEN, *A computational framework for the regularization of adjoint analysis in multiscale pde systems*, Journal of Computational Physics, 195 (2004), pp. 49–89.
- [157] J. QIAN, Y.-T. ZHANG, AND H.-K. ZHAO, *Fast sweeping methods for eikonal equations on triangular meshes*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 83–107.
- [158] S. RAHMATALLA AND C. C. SWAN, *Sparse monolithic compliant mechanisms using continuum structural topology optimization*, International Journal for Numerical Methods in Engineering, 62 (2005), pp. 1579–1605.
- [159] R. SAYE AND J. A. SETHIAN, *A review of level set methods to model interfaces moving under complex physics: Recent challenges and advances*, in Geometric Partial Differential Equations, Part 1, Handbook of Numerical Analysis, 21, Elsevier, 2020, pp. 509–554.

- [160] V. H. SCHULZ, *A Riemannian view on shape optimization*, Found. Comput. Math., 14 (2014), pp. 483–501.
- [161] J. A. SETHIAN, *Fast marching methods*, SIAM review, 41 (1999), pp. 199–235.
- [162] ———, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cambridge university press, 1999.
- [163] J. A. SETHIAN AND A. WIEGMANN, *Structural boundary design via level set and immersed interface methods*, Journal of computational physics, 163 (2000), pp. 489–528.
- [164] K. SHOEMAKE, *Animating rotation with quaternion curves*, SIGGRAPH Comput. Graph., 19 (1985), p. 245–254.
- [165] O. SIGMUND, *On the design of compliant mechanisms using topology optimization*, Journal of Structural Mechanics, 25 (1997), pp. 493–524.
- [166] ———, *A 99 line topology optimization code written in matlab*, Structural and multidisciplinary optimization, 21 (2001), pp. 120–127.
- [167] ———, *Morphology-based black and white filters for topology optimization*, Structural and Multidisciplinary Optimization, 33 (2007), pp. 401–424.
- [168] O. SIGMUND AND K. MAUTE, *Topology optimization approaches*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 1031–1055.
- [169] W. S. SLAUGHTER, *The linearized theory of elasticity*, Springer Science & Business Media, 2012.
- [170] J. SOKOLOWSKI AND A. ZOCHOWSKI, *On the topological derivative in shape optimization*, SIAM Journal on Control and Optimization, 37 (1999), pp. 1251–1272.
- [171] ———, *Topological derivatives for elliptic problems*, Inverse problems, 15 (1999), p. 123.
- [172] J. SOKOLOWSKI AND J.-P. ZOLÉSIO, *Introduction to shape optimization*, Springer, 1992.
- [173] J. STRAIN, *Semi-lagrangian methods for level set equations*, Journal of Computational Physics, 151 (1999), pp. 498–533.
- [174] V. SVERAK, *On optimal shape design*, J. Math. Pures Appl., 72 (1993), pp. 537–551.
- [175] A. TAKEZAWA, S. NISHIWAKI, AND M. KITAMURA, *Shape and topology optimization based on the phase field method and sensitivity analysis*, Journal of Computational Physics, 229 (2010), pp. 2697–2718.
- [176] L. TARTAR, *An introduction to the homogenization method in optimal design*, vol. 1740, Springer, 2000.
- [177] N. P. VAN DIJK, K. MAUTE, M. LANGELAAR, AND F. VAN KEULEN, *Level-set methods for structural topology optimization: a review*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 437–472.
- [178] F. VAN KEULEN, R. T. HAFTKA, AND N. H. KIM, *Review of options for structural design sensitivity analysis. part 1: Linear systems*, Computer methods in applied mechanics and engineering, 194 (2005), pp. 3213–3243.
- [179] C. H. VILLANUEVA AND K. MAUTE, *Density and level set-xfem schemes for topology optimization of 3-d structures*, Computational Mechanics, 54 (2014), pp. 133–150.
- [180] A. WÄCHTER AND L. T. BIEGLER, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical programming, 106 (2006), pp. 25–57.
- [181] F. WANG, B. S. LAZAROV, AND O. SIGMUND, *On projection methods, convergence and robust formulations in topology optimization*, Structural and Multidisciplinary Optimization, 43 (2011), pp. 767–784.
- [182] M. Y. WANG AND X. WANG, *Color level sets: a multi-phase method for structural topology optimization with multiple materials*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 469–496.
- [183] M. Y. WANG, X. WANG, AND D. GUO, *A level set method for structural topology optimization*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.
- [184] M. Y. WANG AND S. ZHOU, *Phase field: a variational method for structural topology optimization*, Comput Model Eng Sci, 6 (2004), pp. 547–566.
- [185] H. ZHAO, *A fast sweeping method for eikonal equations*, Mathematics of computation, 74 (2005), pp. 603–627.
- [186] H.-K. ZHAO, S. OSHER, AND R. FEDKIW, *Fast surface reconstruction using the level set method*, in Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on, IEEE, 2001, pp. 194–201.
- [187] M. ZHOU AND G. ROZVANY, *The coc algorithm, part ii: Topological, geometrical and generalized shape optimization*, Computer methods in applied mechanics and engineering, 89 (1991), pp. 309–336.
- [188] S. ZHOU AND Q. LI, *A variational level set method for the topology optimization of steady-state navier–stokes flow*, Journal of Computational Physics, 227 (2008), pp. 10178–10195.
- [189] S. ZHU, *Effective shape optimization of Laplace eigenvalue problems using domain expressions of Eulerian derivatives*, J. Optim. Theory Appl., 176 (2018), pp. 17–34.