Shape optimization with a level set based mesh evolution method

Grégoire Allaire, Charles Dapogny, Pascal Frey

SHAPE OPTIMIZATION WITH A LEVEL SET BASED MESH EVOLUTION METHOD

G. ALLAIRE[1], C. DAPOGNY[2,4], P. FREY [2,3]

[1] *CMAP, UMR 7641, Ecole Polytechnique, 91128 Palaiseau, France.*
[2] *Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire J.-L. Lions, F-75005 Paris, France.*
[3] *CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.*
[4] *Renault DREAM-DELT'A Guyancourt, France.*

ABSTRACT. In this article, we discuss an approach for geometry and topology optimization of structures which benefits from an accurate description of shapes at each stage of the iterative process - by means of a mesh amenable for mechanical analyses - while retaining the whole versatility of the level set method when it comes to accounting for their evolution. The key ingredients of this method are two operators for switching from a meshed representation of a domain to an implicit one, and conversely; this notably brings into play an algorithm for generating the signed distance function to an arbitrary discrete domain, and a mesh generation algorithm for implicitly-defined geometries.

CONTENTS

1. INTRODUCTION

In the simulation of a free or moving boundary problem driven by a physical motion, one usually has to reconcile numerical accuracy with robustness: the more faithful the representation of the tracked boundary, the more accurate the computation of the motion (i.e. the velocity field driving the motion), and unfortunately, the more tedious the numerical implementation. This issue is especially critical in shape optimization which features problems where the changes in geometry and topology of shapes in the course of the evolution are often dramatic.

Roughly speaking, in the field of shape and topology optimization, three main classes of techniques can be distinguished, depending on the description of shapes they involve:

- *Density-based methods*, such as the SIMP method [17] or the homogenization method [3, 16], transform the problem of finding the optimal shape $\Omega \subset \mathbb{R}^d$ with respect to a mechanical criterion $J(\Omega)$ into that of finding the optimal density function $\rho : D \rightarrow [0, 1]$ of a mixture of material and void inside a fixed working domain D. The shape optimization problem has to be translated into this rather different setting, but the main difficulties are the absence of a clear definition of the shape boundary and the penalization process which, in the end, should deliver a 'classical' shape without intermediate densities.

- *Eulerian methods*, such as the phase field method [18], or the level set method [11, 54, 61, 68] account for shapes in an implicit way; for instance, in the latter case, a large, fixed working domain D, meshed once and for all is introduced, and a shape $\Omega \subset D$ is described in terms of a scalar function $\phi : D \rightarrow \mathbb{R}$ whose negative subdomain matches with Ω. Finite element analyses cannot be performed directly on Ω since it is not meshed exactly, and approximations have to be made to trade mechanical problems posed on Ω for problems posed on D. The most notorious of them is the so-called Ersatz material approach, which consists in filling the 'void' $D \setminus \overline{\Omega}$ with a very soft material to avoid degeneracy in the stiffness matrix (however, alternatives exist, which are based on e.g. the immersed boundary method [61], or the XFEM method [36, 48]).

- *Lagrangian methods* are perhaps the most natural ones and date back to the early hours of computational structural optimization [57, 71]; shapes are represented by means of a computational mesh (or a CAD model [19]), which enables accurate mechanical analyses. The general drawback of this class of methods lies in that this mesh (or whatever explicit representation of shapes is used) has to be updated in the course of the optimization process, which is a notoriously difficult operation, especially in $3d$. Note that there is still ongoing research in this direction [13, 26, 51].

Of course, this rough classification ignores the numerous particular instances of each category of methods and combinations between them (see the recent review papers [31, 33] for a stronger emphasis on level-set based structural optimization).

In the present paper, we describe in details our work, briefly announced in [6, 7], and propose a shape optimization strategy which benefits from the flexibility of the level set method for tracking evolution of shapes, including topological changes, while enjoying an exact, meshed description of shapes. In a nutshell, our approach relies on a level set description of a shape under evolution as a guide for driving the usual mesh operators (edge split, edge collapse, etc...) involved in the update of its mesh. It can be regarded as a systematic and robust way of modifying and controlling the level of detail of this mesh. While they rely on similar mesh operations, Lagrangian methods such as the one presented in [26] generally prove more combinatorial (especially when it comes to the case of three space dimensions), since they use heuristics to avoid producing an invalid mesh or to account for topological changes in the shape.

Admittedly, the idea of combining an implicit domain evolution method with an explicit type of shape representation is not new: in the two-dimensional work [70], the evolution of shapes is tracked on a triangular mesh $\mathcal{T}$ of a working domain D owing to the level set method, and at each iteration of the process, an exact mesh of the current shape Ω is obtained by relocating vertices of $\mathcal{T}$ onto $\partial\Omega$. In [69], a similar strategy is applied for dealing with the motion of shapes; a computational mesh for any shape Ω arising during the process is then constructed by first identifying the intersection points of the implicitly-defined boundary $\partial\Omega$ with the edges of the computational mesh $\mathcal{T}$ of D, then using them as an input for a Delaunay-based mesh generation algorithm. Last but not least, let us mention the work in [56] (Chap. 5), taken over in [45], in which the evolution of shapes is dealt with by using the level set method on a finite difference grid of the working domain D, and an original meshing algorithm for implicit geometries is used to get an exact representation of shapes. The precisely calculated shape gradient must then be projected back to the Cartesian grid of D to close the loop.

Our method has something in common with this last work: a computational domain D is defined, and is equipped with an unstructured mesh which is modified from one iteration of the algorithm to the next, in such a way that any shape Ω arising in the course of the process is *explicitly discretized* in this mesh - i.e. the vertices, edges, faces (and tetrahedra in $3d$) of a mesh of Ω also exist as elements of the ambient mesh of D. In such a configuration, we shall also say that (a mesh of) Ω exists as a *submesh* of that of D. This

kind of representation allows for accurate finite element analyses, held on a well-defined, high-quality mesh of Ω (which is possibly adapted to an error estimate for the mechanical problem at stake), and lends itself to the use of the level set method in an unstructured mesh framework, to account for large shape deformations (including possible topological changes). It relies crucially on efficient algorithms for moving back and forth, from a situation where a shape Ω is known as a submesh of the computational mesh of D to a level set description of Ω on a (unstructured) mesh of D.

This strategy presents several attractive assets; first, no projection between different meshes is needed between the computation of a descent direction for the considered objective function of the domain (which occurs when the shape is in a meshed description), and the further deformation of the shape (which is carried out using the level set method). Most importantly, the proposed method does not pose any theoretical obstruction to the extension from the two-dimensional case to the three-dimensional one (even if it is then considerably more tedious to implement). This is an important and non trivial feature insofar as meshing algorithms are involved, and meshing issues (e.g. Delaunay-based mesh generation) are well-known to be far more difficult to deal with in $3d$ than in $2d$. The only *mesh generation* operation involved in our strategy is a purely logical (thus very robust) one, and most of the difficulty of the problem is transferred to a *remeshing*[1] problem, which always starts from an existing - possibly very ill-shaped mesh - and proceeds 'in the best possible way'.

In this article, we are mostly interested in the three-dimensional setting; consequently, most of the descriptions will be held bearing this case in mind (especially as far as meshing issues are concerned), except when the $2d$ and $3d$ settings are completely equivalent. The proposed approach relies on meshing algorithms which may find other applications (like e.g. in the field of fluid-structure interaction) and have therefore been published in [29], independently from a particular physical or mechanical setting. The novelty of the present paper lies therefore in the interaction of these meshing devices with other level set type algorithms and shape optimization techniques.

This article is organized as follows. The next section presents the model linearized elasticity problem and the basic material from shape-sensitivity analysis (based on Hadamard's method) involved in the gradient optimization algorithm. Section 3 is the central section of the paper and describes the two different representations of shapes used in our method - namely the level set one, and the meshed one - as well as the algorithms for switching from one of these representations to the other. Then, Section 4 describes how the velocity field driving the motion of shapes is computed and how the level set method is used to account for this motion. The global mesh evolution strategy is summed up in Section 5 and several numerical examples are discussed in Section 6. Eventually, we draw some conclusions around the present study in Section 7 and outline a few possible topics for future work.

2. A MODEL PROBLEM IN SHAPE OPTIMIZATION OF ELASTIC STRUCTURES

In this article, we are interested in *shapes*, that is, bounded open sets $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$ in our applications) with at least Lipschitz regularity, filled with a linear isotropic elastic material with Hooke's law A:

$$\forall \xi \in \mathcal{S}(\mathbb{R}^d), \ A\xi = 2\mu\xi + \lambda\mathrm{tr}(\xi),$$

where $\mathcal{S}(\mathbb{R}^d)$ stands for the set of real $d \times d$ symmetric matrices, and λ and μ are the Lamé coefficients of the material, which satisfy $\mu > 0$ and $\lambda + 2\mu/d > 0$. These shapes are clamped on a part $\Gamma_D \subset \partial\Omega$ of their boundaries. They are submitted to body forces f, as well as to traction loads g, applied on a part $\Gamma_N \subset \partial\Omega$ disjoint from Γ_D. The remaining, traction-free region $\Gamma := \partial\Omega \setminus (\Gamma_D \cup \Gamma_N)$ is called the *free boundary* of Ω.

Provided $f \in L^2(\mathbb{R}^d)^d$, $g \in H^1(\mathbb{R}^d)^d$, and that $\Gamma_D \neq \emptyset$ (unless an equilibrium condition between f and g is fulfilled), the displacement of a shape Ω is the unique solution $u_\Omega \in H^1(\Omega)^d$ to the linear elasticity system:

$$(1) \qquad \begin{cases} -\mathrm{div}(Ae(u)) &= f &\text{in } \Omega \\ u &= 0 &\text{on } \Gamma_D \\ Ae(u)n &= g &\text{on } \Gamma_N \\ Ae(u)n &= 0 &\text{on } \Gamma \end{cases},$$

---

[1]Depending on the authors, this term may either refer to the operation of *creating* a whole new mesh, or to that of *modifying* an existing one by means of local mesh operations. In this paper, the latter meaning is retained.

where $e(u) := \left(\nabla u^T + \nabla u\right)/2$ is the linearized strain tensor, and n is the unit normal vector to $\partial\Omega$ (pointing outward Ω).

Our purpose is to minimize a given functional $J(\Omega)$ of the domain. This classically demands some knowledge about the derivatives of J, hence the need to account for variations of shapes. In this perspective, we rely on Hadamard's boundary variation method [2, 47, 52, 63]: variations of a smooth shape $\Omega \subset \mathbb{R}^d$ of the form $\Omega_\theta := (I + \theta)(\Omega)$ are considered, for $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, $||\theta||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)}< 1$. It is indeed well-known that under such conditions, $(I + \theta)$ is a Lipschitz diffeomorphism of $\mathbb{R}^d$. The induced notion of shape differentiation is then the following:

Definition 1. *A real-valued function $J(\Omega)$ of the domain is shape differentiable at a particular shape Ω if the underlying function $\theta \mapsto J((I+\theta)(\Omega))$ from $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ into $\mathbb{R}$ is Fréchet differentiable at $\theta = 0$. The associated derivative $J'(\Omega)$ is called the shape derivative of J at Ω, and the following asymptotic expansion holds in the neighborhood of $0 \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$:*

$$(2) \qquad J(\Omega_\theta) = J(\Omega) + J'(\Omega)(\theta) + o(\theta), \quad where \quad \frac{o(\theta)}{||\theta||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)}} \xrightarrow{\theta\to 0} 0.$$

Let us now specify the setting of our study. The parts Γ_D and Γ_N of the boundaries of shapes where they are respectively clamped and submitted to surface loads are given *a priori*, and are not subject to optimization. The minimization of $J(\Omega)$ is thus investigated over the set $\mathcal{U}_{ad}$ of *admissible shapes* defined as:

$$\mathcal{U}_{ad} = \left\{ \Omega \subset \mathbb{R}^d \text{ is an open, Lipschitz bounded set, } \Gamma_D \cup \Gamma_N \subset \partial\Omega \right\}.$$

The corresponding set for *admissible variations* of shapes is:

$$\Theta_{ad} = \left\{ \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \ \theta = 0 \text{ on } \Gamma_D \cup \Gamma_N \right\}.$$

Throughout this chapter, we shall consider integral functionals $J(\Omega)$, which bring into play the solution u_Ω to the linear elasticity system (1). The shape derivative of such *state-constrained* functionals can be computed thanks to techniques from optimal control theory (such as Céa's method [23]).

To set ideas, let us recall a classical result, devoted to functionals of the form (see e.g. [11] for details):

$$(3) \qquad \forall\Omega \in \mathcal{U}_{ad}, \ J(\Omega) = \int_\Omega j(x, u_\Omega(x)) \, dx + \int_{\Gamma\cup\Gamma_N} k(x, u_\Omega(x)) \, ds,$$

where $j, k : \mathbb{R}^d_x \times \mathbb{R}^d_u \to \mathbb{R}$ are two smooth functions satisfying adequate growth conditions (we shall meet several instances of such objective functions in Section 6).

Theorem 1. *Provided f, g and Ω are smooth enough, the function $J(\Omega)$ defined by (3) is shape differentiable at any $\Omega \in \mathcal{U}_{ad}$, and its shape derivative reads:*

$$(4) \quad \forall\theta \in \Theta_{ad}, \ J'(\Omega)(\theta) = \int_\Gamma \left(j(x, u_\Omega) + Ae(u_\Omega) : e(p_\Omega) - f \cdot p_\Omega + \frac{\partial}{\partial n}(k(x, u_\Omega)) + \kappa k(x, u_\Omega) \right) \theta \cdot n \, ds,$$

where κ is the mean curvature of $\partial\Omega$ (oriented in the sense that $\kappa(x)$ is positive when Ω is locally convex around x), and $p_\Omega \in H^1(\Omega)^d$ is the adjoint state, unique solution to:

$$(5) \qquad \begin{cases} -\mathrm{div}(Ae(p)) & = & -j'(x, u_\Omega) & in \ \Omega \\ p & = & 0 & on \ \Gamma_D \\ Ae(p)n & = & -k'(x, u_\Omega) & on \ \Gamma \cup \Gamma_N \end{cases}.$$

More generally, in good agreement with the structure theorem (see [32], Th 9.3.6), the shape derivatives of all the considered functionals $J(\Omega)$ in this paper will turn out to be of the form:

$$(6) \qquad \forall\theta \in \Theta_{ad}, \ J'(\Omega)(\theta) = \int_\Gamma w_\Omega \, \theta \cdot n \, ds,$$

for a certain scalar field w_Ω on Γ, which depends on u_Ω, and possibly on some *adjoint state* p_Ω. A descent direction θ for J is then easily revealed as $-w_\Omega n$, since letting

$$(7) \qquad \theta = -t \, w_\Omega n$$

in (2) yields, for small $t > 0$ (provided $w_\Omega \neq 0$):

$$J(\Omega_{t\theta}) = J(\Omega) - t \int_\Gamma w_\Omega^2 \, ds + o(t) < J(\Omega).$$

Actually, for both theoretical and numerical reasons, one cannot take directly (7) as a descent direction; we shall come back to this issue in Section 4.3.

Remark 1. We have hitherto been discussing the unconstrained minimization of a function $J(\Omega)$ over $\mathcal{U}_{ad}$. For the sake of simplicity, in this paper, we shall limit ourselves with imposing a volume constraint on shapes, to be enforced by trading $J(\Omega)$ for a weighted sum $\mathcal{L}(\Omega)$ of $J(\Omega)$ and the volume of shapes $\mathrm{Vol}(\Omega)$, so that the problem boils down to the following constraint-free problem:

$$(8) \qquad \min_{\Omega \in \mathcal{U}_{ad}} \mathcal{L}(\Omega), \ \ \mathcal{L}(\Omega) := J(\Omega) + \ell \, \mathrm{Vol}(\Omega),$$

where ℓ is a fixed Lagrange multiplier. Note that this very rough understanding of constraints already contains some degree of generality, since many efficient optimization algorithms (e.g. the augmented Lagrangian method) impose constraints by using formulations of the form (8) in combination with an update strategy for the Lagrange multiplier ℓ.

3. Two complementary ways for representing shapes

From now on, let D be a fixed, large computational domain that encloses all the considered shapes. The central point of the proposed method consists in juggling with two different ways for describing a shape $\Omega \subset D$ during the optimization process (see Figure 1), using alternatively one or the other depending on the nature of the ongoing operation:

- *The level set description:* Ω is implicitly defined by a scalar 'level set' function $\phi : D \to \mathbb{R}$, in the sense that the following holds:

$$(9) \qquad \forall x \in D, \ \begin{cases} \phi(x) < 0 & \text{if } x \in \Omega \\ \phi(x) = 0 & \text{if } x \in \partial\Omega \\ \phi(x) > 0 & \text{if } x \in {}^c\overline{\Omega} \end{cases}.$$

  From the numerical standpoint, ϕ is discretized as a $\mathbb{P}^1$ Lagrange finite element function on a (simplicial) mesh of D. As we shall recall in Section 4, this way of representing shapes is particularly well-suited when its comes to tracking their evolutions.
- *The meshed description:* the whole computational domain D is equipped with a simplicial (conforming) mesh $\mathcal{T}_\Omega$, which encloses a mesh $\mathcal{T}'_\Omega$ of Ω as a *submesh*, i.e. the elements (points, edges, faces, and tetrahedra in 3d) of $\mathcal{T}'_\Omega$ also exist as elements of the larger mesh $\mathcal{T}_\Omega$.

  This description of Ω is convenient when it comes to performing mechanical computations on it (e.g. using the finite element method): to achieve this, one only has to forget the 'exterior' part $\mathcal{T}_\Omega \setminus \mathcal{T}'_\Omega$ of $\mathcal{T}_\Omega$, i.e. that corresponding to $D \setminus \Omega$, and retain only the computational mesh $\mathcal{T}'_\Omega$ of Ω.

  At this point, one may question our choice of systematically meshing a shape Ω *together* with its complementary part $D \setminus \Omega$, but the need to do so will become apparent in the next sections.

Let us now describe the two operators at our disposal for switching from one representation to the other.

3.1. **Generating the signed distance function to a discrete domain.**

The first operation under scrutiny consists in generating a level set function for a domain $\Omega \subset \mathbb{R}^d$, at the vertices of a mesh $\mathcal{T}_\Omega$ of D in which Ω is explicitly discretized. Such a function is computed as the *signed distance function* d_Ω to Ω, which is defined by:

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

where $d(., \partial\Omega)$ is the Euclidean distance function to $\partial\Omega$. Indeed, since [25], several properties of the signed distance function - the most crucial of them being that its gradient is of unit norm wherever it is defined,
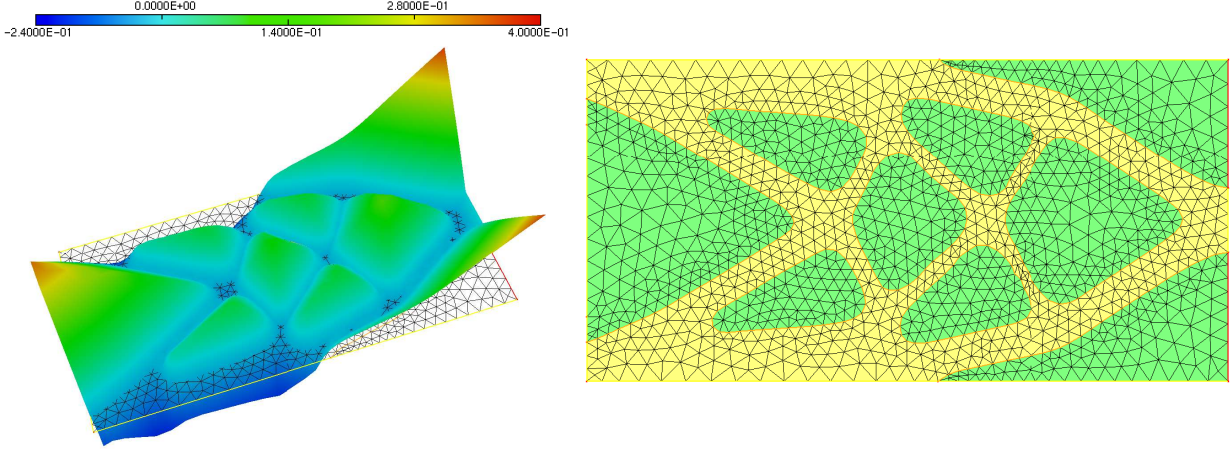
FIGURE 1. *(Left)* graph of a level set function ϕ associated to a shape Ω; *(right)* a corresponding meshed description: the whole computational box D is equipped with a mesh $\mathcal{T}_\Omega$ (composed of the yellow and green elements), and the submesh $\mathcal{T}'_\Omega$ is composed of the yellow triangles.

i.e. $|\nabla d_\Omega| = 1$ a.e. on $\mathbb{R}^d$ - have proved to tremendously increase the numerical accuracy and stability of computations performed within a process making use of the level set method.

It is well-known [14], [53], [60] that d_Ω can be obtained as the stationary state of the *unsteady Eikonal equation*:

$$(10) \qquad \begin{cases} \frac{\partial \phi}{\partial t}(t,x) + \mathrm{sgn}(\phi_0)\,(|\nabla \phi| - 1) = 0 & \text{for } (t,x) \in (0,\infty) \times \mathbb{R}^d \\ \phi(t=0,x) = \phi_0(x) & \text{for } x \in \mathbb{R}^d \end{cases},$$

where ϕ_0 is *any* level set function associated to Ω. Note that such a function is easily generated in practice, e.g. by defining $\phi_0(x)$ as the exact signed distance function to Ω at 'close' vertices $x \in \mathcal{T}_\Omega$ to $\partial \Omega$ (which is then inexpensive), and with an arbitrarily large value at the remaining points of $\mathcal{T}_\Omega$.

Since computing the solution of (10) proves useful in many problems with applications outside from the field of structural optimization, we proposed a numerical scheme which was independently published in [30]. It relies on an explicit formula [14] for the solution to (10). It is implemented as an iterative algorithm on unstructured meshes which 'straightens up' an initial level set function ϕ_0 for Ω into the signed distance function, updating the values of the function from a neighbourhood of $\partial \Omega$ to further regions.

Remark 2. The above assumption, whereby Ω should be explicitly discretized in the computational mesh of D is not mandatory for this operation: Ω could actually be supplied via *any* mesh, independently from that of D; see [30] for details.

3.2. **Meshing the negative subdomain of a scalar function.**

The second operation of interest assumes the data of a simplicial mesh $\mathcal{T}$ of D and a level set function ϕ, discretized at the vertices of $\mathcal{T}$, associated to a domain $\Omega \subset D$. The aim is to modify $\mathcal{T}$ into a new *well-shaped and adapted* mesh $\mathcal{T}_\Omega$ of D in which Ω is explicitly discretized.

We impose two additional features to the new mesh $\mathcal{T}_\Omega$:

- The mesh $\mathcal{T}_\Omega$ should be *well-shaped*, in terms of the *qualities* of its element. Depending on the desired application, there are many different ways for measuring the quality of an element. Since we plan to perform mechanical analyses on $\mathcal{T}_\Omega$, in particular by using the finite element method, a natural choice would consist in evaluating a simplex K in terms of its *eccentricity* $\sigma_K = \rho_K/h_K$ (where ρ_K is the inradius of K, and h_K is its diameter), which is a quantity involved in many finite element a priori estimates (see e.g. [27]). We rather rely on another function $Q(K)$ of a simplex K with edges

$e_1, ..., e_{d(d+1)/2}$, whose meaning is close to that of σ_K, but which shows a better numerical ability in discriminating 'good' from 'average', or 'bad' elements:

$$Q(K) = \alpha \frac{\text{Vol}(K)}{\left(\sum\limits_{i=1}^{d(d+1)/2} |e_i|^2 \right)^{\frac{d}{2}}},$$

where $\alpha > 0$ is a normalization factor, defined so that $Q(K) = 1$ if K is a regular simplex.

- The submesh $\mathcal{T}'_\Omega$ of $\mathcal{T}_\Omega$ should be *adapted* to the geometrical features of Ω, in particular show smaller elements around the regions of $\partial\Omega$ where curvature is high. This requirement might seem a bit loose at first glance, since in our framework, ϕ and Ω are only known in a discrete way - and are respectively a piecewise linear function and a polyhedral domain (thus, strictly speaking, there is no such thing as curvature as far as Ω is concerned). Actually, for a number of reasons, it proves convenient to rely consistently on a continuous geometric model for Ω; such a model can be inferred from the datum of any mesh of Ω, by first inferring local geometric information about this continuous model from the discrete data at hand (for instance, normal vectors at the vertices of the mesh can be computed from the normal vectors the the surrounding surface triangles), and then by following rules to generate a local portion of an underlying continuous surface to $\partial\Omega$ - hereafter denoted as Γ - from any given discrete surface triangle T of $\partial\Omega$. In our setting, as suggested in [67], this piece of surface is parametrized as a cubic Bézier patch $\sigma : T \to \Gamma$, which interpolates the three vertices of T, together with the three associated normal vectors. This local model serves then as a guide when it comes to introducing new points on $\partial\Omega$, and results in simple predicates over the vertices and normal vectors of the surface mesh when it comes to measuring whether such or such operation degrades too much the geometrical features of Ω.

Modifying $\mathcal{T}$ into such a mesh $\mathcal{T}_\Omega$ is achieved within two steps, which we now briefly outline: at first, a mesh $\mathcal{T}_{temp}$ of D is obtained, in which Ω is explicitly discretized, but which may be very ill-shaped, or may be a poor representative of the geometry of Ω. In a second step, this intermediate mesh $\mathcal{T}_{temp}$ is remeshed into a high-quality mesh $\mathcal{T}_\Omega$, which is a fine representative of Ω. For the many technical details the reader is refered to [28, 29]. Remark that these meshing operations have many other applications, as described in [29].

3.2.1. *Step* 1: *discretization of the* 0 *level set of a scalar function into a simplicial mesh.*

If no particular attention is paid to the qualities of its elements, obtaining a mesh $\mathcal{T}_{temp}$ of D in which Ω is explicitly discretized is a fairly easy matter: we use a *marching tetrahedra* approach [39] - a variant of the well-known marching cubes algorithm which assumes a Cartesian computational support [49].

The simplices of $\mathcal{T}$ intersecting $\partial\Omega = \{x \in D, \ \phi(x) = 0\}$ are exactly those bearing at least two vertices where ϕ takes different signs. For any such simplex $K \in \mathcal{T}$, as $\phi|_K$ is linear, $\partial\Omega \cap K$ is a portion of plane passing through those points m_i of the edges of K where ϕ vanishes. Once the positions of these points have been computed, depending on the relative signs of ϕ at the vertices of K, a pattern is chosen for splitting K into several simplices in such a way that a triangulation of $\partial\Omega \cap K$ explicitly appears (see Figure 2 for a three-dimensional example). There are two such patterns in two dimensions, four in three dimensions, up to rotations of the configuration.

This step is unfortunately very likely to produce a severely ill-shaped mesh $\mathcal{T}_{temp}$, with very small or nearly degenerate elements, since the intersections of the simplices $K \in \mathcal{T}$ with $\partial\Omega$ are arbitrary (for instance, in the configuration of Figure 2, if the portion of plane $\partial\Omega \cap K$ lies very close to the vertex a_3, the new tetrahedron $a_3 m_0 m_1 m_2$ will be too small and $a_0 m_0 m_1 m_2$ will be almost flat).

Nevertheless, let us note that this purely logical step will be the only mesh *generation* operation involved in the mesh evolution method at stake in this paper. It is very robust in delivering *valid* simplicial meshes. The remaining meshing ingredients, whose descriptions follow, only consider, modify and deliver valid meshes, doing their utmost in increasing their qualities. This commitment in reducing to a minimum the true mesh *generation* component in our approach is the main reason for its robustness.
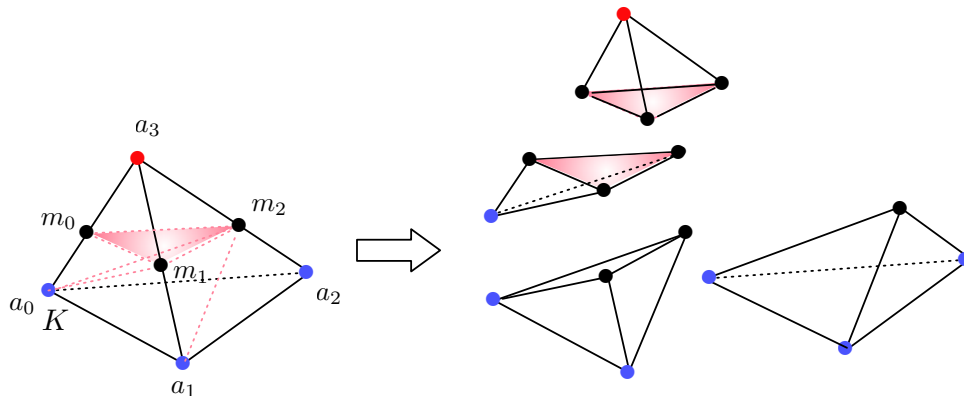
FIGURE 2. One possible pattern for splitting a simplex K so that the 0 level set (red face) of a linear function ϕ in K explicitly appears in the resulting decomposition. Here, ϕ takes a positive value at the red vertex of K, and negative ones at the blue vertices.

3.2.2. *Step 2: local mesh modifications for quality and geometric approximation improvements.*

We are now left with the problem of remeshing a possibly ill-shaped simplicial mesh $\mathcal{T}_{temp}$ of D, enclosing an explicit discretization of Ω (which may be poor as a geometric approximation of the continuous underlying model).

To achieve this, we rely on the four usual local mesh modification operators (see [40], or [28, 29] for more details around the actual implementation), which are briefly described hereafter (see Figure 3). Note that, in our application, each one of them exists under two different forms depending on whether it is applied to a surface configuration, or to a completely internal one.

(a) *Edge split:* an edge pq of $\mathcal{T}_{temp}$ which is 'too long' is split by introducing a new vertex m in the mesh, and replacing pq by pm and qm, updating the connectivities of the mesh accordingly. An edge may be deemed 'too long' if it is too long with respect to a user-defined size prescription, or if it entails too large a gap between $\partial\Omega$ and the underlying continuous surface Γ. The new vertex m is inserted either on Γ if pq is an edge of $\partial\Omega$, or as the midpoint of pq if it is an internal edge.

(b) *Edge collapse:* the two endpoints of a 'too short' edge pq of $\mathcal{T}_{temp}$ are merged. This operator should be cautiously monitored: not only is it likely to degrade the quality of the representation of the geometrical features of Ω, but it may also invalidate elements of the mesh (i.e. cause overlappings), or provoke topologically invalid (e.g. non manifold) surface configurations.

(c) *Edge swap:* An edge pq is removed in $\mathcal{T}_{temp}$, and the connectivities of the mesh are updated accordingly. The modus operandi of this operator is easy to apprehend when it is applied to $2d$ or $3d$ surface configurations: in this case, a configuration of two triangles $T_1 = apq$ and $T_2 = bpq$ sharing the common edge pq is simply replaced by the alternate configuration of triangles $\widetilde{T_1} = pab$ and $\widetilde{T_2} = qab$, sharing the edge ab (see Figure 3 (c)). However, it becomes much more combinatorial and tedious in $3d$ (and the designation 'edge swap' may seem a little be improper). In this case, the whole 'shell' of tetrahedra sharing the edge pq has to be reconnected, and there are many possible ways to do so, not all of them resulting in a valid configuration of suitable quality; see Figure 4 and [34, 42], or [29] for further details. In either case, this operator too should be carefully controlled, as it may invalidate $\mathcal{T}_{temp}$, or degrade the geometric features of $\partial\Omega$.

(d) *Vertex relocation:* A vertex $p \in \mathcal{T}_{temp}$ is relocated to an *improving position* $\widetilde{p}$, leaving the connectivities of $\mathcal{T}$ unchanged. The choice as for the improving position $\widetilde{p}$ depends on whether p belongs to $\partial\Omega$ or it is an internal vertex. In the former case, $\widetilde{p}$ should lie on the continuous model Γ associated to $\partial\Omega$, whereas in the latter case, it is simply chosen as the centroid of the simplices sharing p as a vertex (see however [40, 38] for other possibilities as for the relocation position).
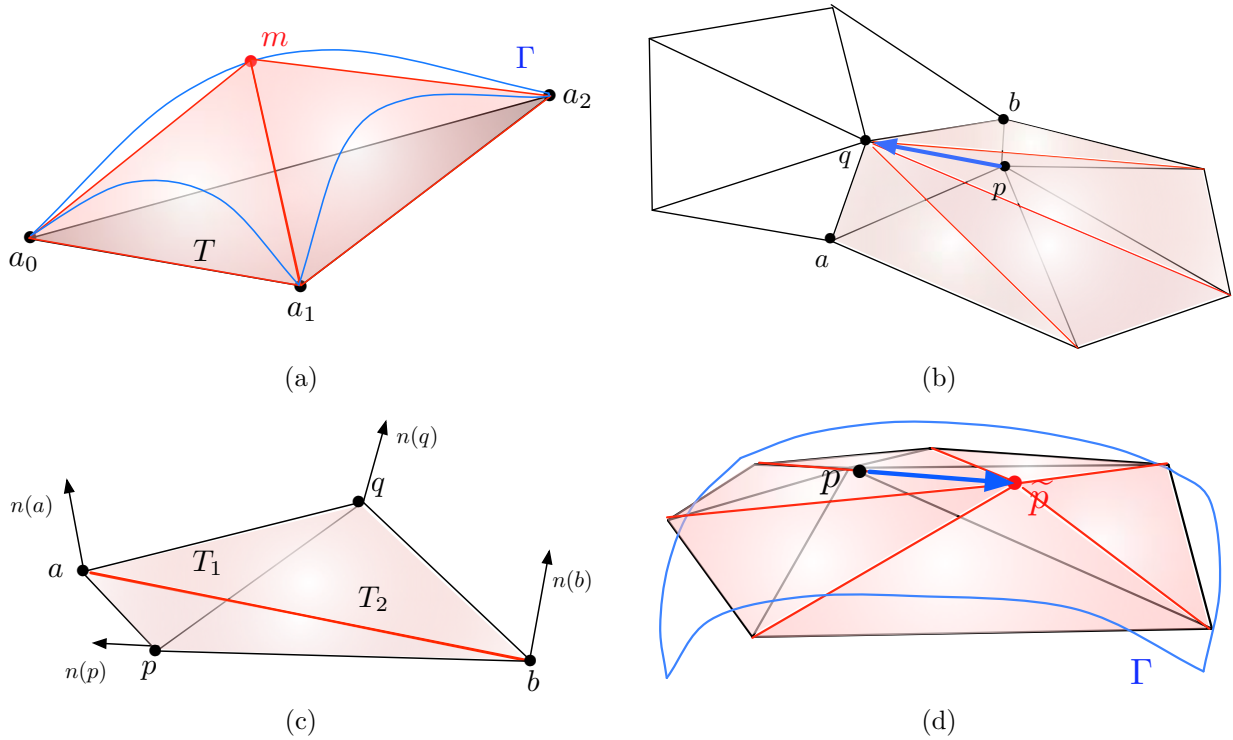
8

FIGURE 3. The four local remeshing operators, applied to a boundary configuration. In all four pictures, the initial configuration is shown in black, and the final one in red: *(a)* split of an edge $a_0 a_2$ of a surface triangle T, *(b)* collapse of a boundary edge pq; *(c)* swap of a boundary edge pq; *(d)* relocation of a vertex p to an improving position $\widetilde{p}$.
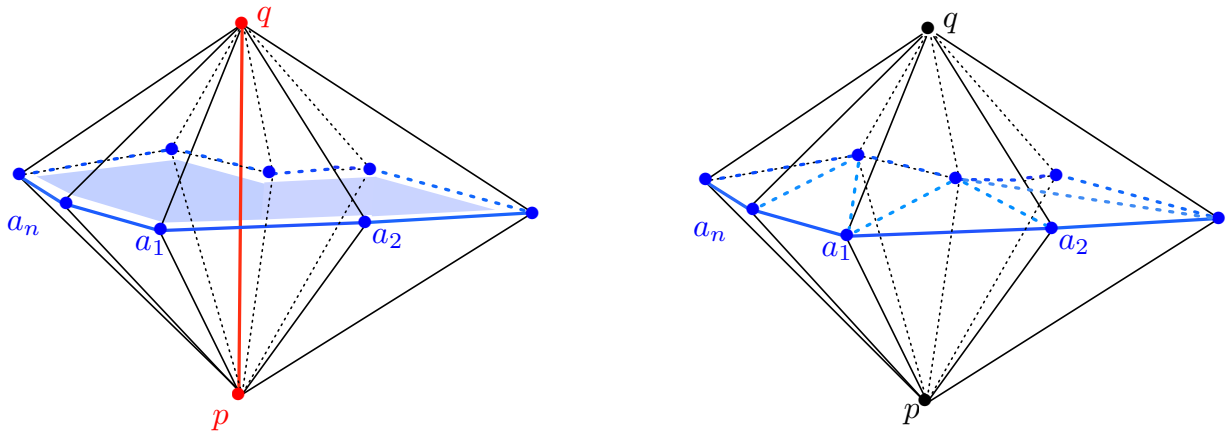

FIGURE 4. Swap of an edge pq in $3d$. *(Left)* The pseudo-polygon consisting of the vertices of the removed tetrahedra in the 'shell' of pq is enumerated (in light blue), and triangulated before being reconnected to p and q; *(right)*: one of the many possibilities.

These local remeshing operators serve different purposes: the first two ones (a) and (b) are mainly 'sampling operators', insofar as they make it possible to reach a desired element density in terms of a user-defined size prescription, or of geometric approximation concerns, that is, a suitable level of detail. The last two ones (c) and (d) are essentially quality improvement operators.

The way to combine these four operators - however completely heuristic an issue - turns out to be at least as important as their individual performances in a remeshing algorithm. Without delving into details, here is the outline of the global strategy which showed the best efficiency in our study.

(1) In a first stage, operations are focused on modifying $\mathcal{T}_{temp}$ into a 'geometric mesh' $\widetilde{\mathcal{T}_{temp}}$ of D, with respect to Ω: $\widetilde{\mathcal{T}_{temp}}$ may still be very ill-shaped, but it encloses a discretization of a close approximation to the continuous geometric model for Ω. This stage mostly involves edge split and collapse operations.

(2) A *size map* $h : D \to \mathbb{R}$ is computed (it is actually stored at the vertices of $\widetilde{\mathcal{T}_{temp}}$), which describes the local desired size features for remeshing, based on curvature estimates at the vertices of $\partial\Omega$, and taking into account user-defined bounds for the minimal and maximal authorized edge lengths.

(3) The intermediate mesh $\widetilde{\mathcal{T}_{temp}}$ is modified into the high-quality mesh $\mathcal{T}_\Omega$: edge splits and collapses are performed to reach the size feature enclosed in the size map h; at first, only the configurations which are 'very much' deviant from the size prescription are considered, then the criteria become increasingly strict. These operations are intertwined with massive uses of edge swaps and vertex relocations, whenever they help in improving the overall quality of the mesh.

Remark 3. This remeshing algorithm can serve two additional purposes, illustrations of which are provided in the numerical examples of Section 6.

- When using the 'classical' level set method of [11, 68] on a fixed mesh of D, it allows the user to obtain a mesh of the resulting optimal shape. From this point, the user could carry on a new shape optimization procedure using the method presented here, hopefully leading to a more accurate result.
- It can produce a high-quality mesh of the final shape of the presented shape optimization process, as a first step towards its post-processing (e.g. in reverse engineering, or when it comes to converting it into a CAD model).

Remark 4. Compared to Lagrangian methods such as the one presented in [26], our remeshing algorithm is simpler in the sense that it does not have to care about topological changes. These latter ones are taken into account in a different algorithmic part of our method, namely in the advection of the level set function, as described in Section 4. Therefore, we do not need to resort to the topological gradient or to combinatorial heuristics (especially in three space dimensions) for handling topological changes in the mesh of the structure.

4. ACCOUNTING FOR SHAPE EVOLUTION

4.1. **A brief reminder of the level set method.**

The evolution of shapes is numerically tracked while they are under implicit representation, by using the level set method [55], originally introduced in the context of shape optimization in [11, 68].

Roughly speaking (see also [53, 60] for more details), let $\Omega(t) \subset \mathbb{R}^d$ be a domain, whose motion over a time period $[0, T]$ is driven by a velocity field $V : [\mathbb{R}^d \to \mathbb{R}^d$. At any time $t \in [0, T]$, let also $\phi(t, .)$ be a level set function associated to $\Omega(t)$. The motion of $\Omega(t)$ is translated in terms of ϕ into the following *level set advection equation*:

$$(11) \qquad \frac{\partial \phi}{\partial t}(t, x) + V(t, x) \cdot \nabla \phi(t, x) = 0 \text{ on } (0, T) \times \mathbb{R}^d.$$

If in addition V is consistently oriented along the normal to $\Omega(t)$, say $V(t, x) = v(t, x)\, n_{\Omega(t)}(x)$, for a certain scalar field v, where $n_{\Omega(t)} = \frac{\nabla \phi(t, .)}{|\nabla \phi(t, .)|}$ denotes (an extension of) the outer unit normal vector to $\partial\Omega(t)$, (11) is best rewritten as a *Hamilton-Jacobi equation*:

$$(12) \qquad \frac{\partial \phi}{\partial t}(t, x) + v(t, x)|\nabla \phi(t, x)| = 0 \text{ on } (0, T) \times \mathbb{R}^d.$$

Since in the present context the time interval $(0, T)$ stands for a 'small' generic time period between two optimization iterates, we shall assume that the velocity does not depend on time or 'freeze' it at its initial value, namely $V(t, x) \approx V(0, x) =: V(x)$ over $[0, T]$, so that (11) becomes a *passive transport equation*:

$$(13) \qquad \frac{\partial \phi}{\partial t}(t, x) + V(x) \cdot \nabla \phi(t, x) = 0 \text{ on } (0, T) \times \mathbb{R}^d.$$

When $V(t,x) = v(t,x) \, n_{\Omega(t)}(x)$ is oriented along the normal to $\Omega(t)$, another possibility consists in freezing only the scalar field v over $[0,T]$, i.e. assuming $v(t,x) \approx v(0,x) =: v(x)$. Thus (12) becomes a *passive* (still non linear) *Hamilton-Jacobi equation*:

$$\text{(14)} \qquad \frac{\partial \phi}{\partial t}(t,x) + v(x)|\nabla \phi(t,x)| = 0 \text{ on } (0,T) \times \mathbb{R}^d,$$

a structure which preserves the information that the velocity has a normal direction to $\partial \Omega(t)$ at any time. Of course, all the above transport equations are equiped with an initial condition

$$\phi(t = 0, x) = \phi_0(x) \quad \text{for } x \in \mathbb{R}^d.$$

4.2. **Resolution of the level set advection equation on an unstructured mesh.**

In the present work we choose to solve (13) by the *method of characteristics*, following an idea of [64]. It has the advantage of being well adapted to unstructured meshes and to be unconditionally stable. The principle of this approach (see [59]) is that, under suitable regularity and growth hypotheses on V and ϕ_0, the unique $\mathcal{C}^1$ solution ϕ to (13) is

$$\text{(15)} \qquad \phi(t,x) = \phi_0\left(X(0,t,x)\right),$$

where $s \mapsto X(s,t,x)$ is the *characteristic curve* of V passing at x at time t, defined as the solution to the ODE:

$$\text{(16)} \qquad \begin{cases} \frac{dX}{ds}(s,t,x) = V(X(s,t,x)) \text{ for } s \in \mathbb{R} \\ X(t,t,x) = x \end{cases},$$

which describes the trajectory of a particle driven be the velocity field V standing at x at time t.

In the numerical setting of this paper, V and ϕ_0 are discretized as $\mathbb{P}^1$ Lagrange finite element functions on a (simplicial) mesh $\mathcal{T}$ of the computational domain D, and an approximation ϕ_T of the solution ϕ to (13) at time $t = T$ is sought under the same form. To achieve this, we simply mimic formula (15): ϕ_T is computed at the vertices of $\mathcal{T}$, using the following formula:

$$\forall \text{ vertex } x \in \mathcal{T}, \ \ \phi_T(x) = \phi_0(\widetilde{X}(0,t,x)),$$

where $\widetilde{X}(0,t,x)$ is a numerical approximation to $X(0,t,x)$, provided by a numerical integration of the ODE (16), e.g. using a first-order Euler's method, or a more accurate Runge-Kutta scheme (see [20] for details about the actual implementation).

4.3. **Computation of a descent direction.**

From a given shape $\Omega \in \mathcal{U}_{ad}$, a descent direction $V_\Omega \in \Theta_{ad}$ for the considered objective function $J(\Omega)$ is computed on a whole mesh $\mathcal{T}_\Omega$ of D which encloses an explicit discretization of Ω.

The generic expression (6) for the shape derivative of J suggests the immediate choice:

$$\text{(17)} \qquad \forall x \in \partial\Omega, \ \ V_\Omega(x) = -w_\Omega(x) \, n(x).$$

As we have seen, w_Ω depends on the solution to one or several systems of the form (1) posed on Ω, which can be accurately solved on the submesh $\mathcal{T}_\Omega'$ of D by using the finite element method. Unfortunately, the choice (17) for a descent direction turns out to be hazardous for two independent reasons:

- Formula (17) makes sense only on $\partial\Omega$, whereas V_Ω should be defined in D or at least in a vicinity of $\partial\Omega$. This feature is imposed by the theoretical requirement that V_Ω should belong to Θ_{ad}, and by the numerical setting of the level set method.
- As exemplified by Theorem 1, the scalar field w_Ω generally depends on (traces of) derivatives of the solution u_Ω to (1) (and possibly on those of the adjoint state p_Ω), which may lack smoothness, both from the theoretical and numerical standpoints. This feature may endanger the numerical stability of the process.

As advocated by [21, 43], an efficient way to address both problems at the same time consists in using as a descent direction the gradient of J associated to a different scalar product from the canonical one of $L^2(\Gamma)$.

More accurately, let $\alpha > 0$ be a small 'extension - regularization' parameter, and let us introduce the functional space

$$H^1_{\Gamma_D \cup \Gamma_N}(D) = \left\{ w \in H^1(D), \ w = 0 \text{ on } \Gamma_D \cup \Gamma_N \right\}.$$

Let also $\widetilde{w} \in H^1_{\Gamma_D \cup \Gamma_N}(D)$ be the unique solution to the variational problem (see [21] for alternative choices):

(18) $\qquad \forall z \in H^1_{\Gamma_D \cup \Gamma_N}(D), \ \int_D (\widetilde{w}z + \alpha \nabla \widetilde{w} \cdot \nabla z) \ dx = J'(\Omega)(zn) \ \left(= \int_\Gamma w_\Omega z \, ds \right).$

Consider now the choice:

(19) $\qquad\qquad\qquad\qquad \forall x \in D, \ \widetilde{V_\Omega}(x) = -\widetilde{w}(x)\, n(x),$

where n stands for (an extension to D of) the normal vector field to $\partial\Omega$. Combining (18) with the asymptotic expansion (2) shows that $\widetilde{V_\Omega}$ is also a descent direction for J. However, $\widetilde{V_\Omega}$ intrinsically enjoys more regularity than V_Ω owing to the classical regularity theory for elliptic equations, and is inherently defined on the whole domain D.

In the numerical setting, $\widetilde{w}$ can be easily computed by solving (18) with the classical finite element method, performed on mesh $\mathcal{T}_\Omega$, after w_Ω has been computed. Note that the discretization of the right-hand side in (18) is straightforward since the computational mesh $\mathcal{T}_\Omega$ encloses an explicit discretization of $\partial\Omega$. The (vector) velocity field $\widetilde{V_\Omega}$ is eventually derived once Ω has been associated a level set function ϕ, by using the usual extension of the normal vector field $n = \frac{\nabla\phi}{|\nabla\phi|}$. In practice we use the value of α given by $4h^2$ where h is the minimal size of elements (prescribed in the remeshing process).

5. The global algorithm

Gathering the ingredients of the previous sections, we are now in a position to describe our general strategy for handling mesh evolution in the context of shape optimization (see Figure 5 for an illustration).

Start with an initial shape Ω^0, and a simplicial mesh $\mathcal{T}_{\Omega^0}$ of D in which Ω^0 is explicitly discretized.

For $k = 0, \dots$ **until convergence**, the current shape Ω^k is known via a mesh $\mathcal{T}_{\Omega^k}$ of D, a submesh $\mathcal{T}'_{\Omega^k}$ of which is a mesh of Ω^k.

(1) Compute the value of the scalar field w_{Ω^k} appearing in the shape derivative of the considered functional (6). This may involve one, or several finite element analyses for solving the state (1) and (possibly) adjoint systems, to be held on the part $\mathcal{T}'_{\Omega^k}$ of the mesh $\mathcal{T}_{\Omega^k}$ corresponding to Ω^k. The quantity w_{Ω^k} is defined only on $\partial\Omega^k$, i.e. in the numerically setting, on the discretization of $\partial\Omega^k$ which explicitly appears in both $\mathcal{T}_{\Omega^k}$ and $\mathcal{T}'_{\Omega^k}$.

(2) Generate the signed distance function d_{Ω^k} to Ω^k on the whole mesh $\mathcal{T}_{\Omega^k}$ of D.

(3) Extend w_{Ω^k} to a vector field V_{Ω^k} defined on the whole mesh $\mathcal{T}_{\Omega^k}$ of D, by using formula (19).

(4) Choose a descent step $\tau^k > 0$, and solve the following level set advection equation on $\mathcal{T}_{\Omega^k}$:

$$\begin{cases} \frac{\partial\phi}{\partial t}(t,x) + V_{\Omega^k}(x) \cdot \nabla\phi(t,x) & \text{for } (t,x) \in (0,\tau^k) \times D \\ \phi(t=0,x) = d_{\Omega^k}(x) & \text{for } x \in D \end{cases}.$$

This produces a new level set function $\phi^{k+1} := \phi(\tau^k,.)$ associated to the new shape Ω^{k+1}.

(5) Obtain the meshed representation of Ω^{k+1} by using the algorithm of Section 3.2 from the set of data $(\mathcal{T}_{\Omega^k}, \phi^{k+1})$: a new mesh $\mathcal{T}_{\Omega^{k+1}}$ of D is produced, which encloses a mesh $\mathcal{T}'_{\Omega^{k+1}}$ of Ω^{k+1}.

(6) Evaluate $J(\Omega^{k+1})$. If $J(\Omega^{k+1}) < J(\Omega^k)$, Ω^{k+1} is retained as the new shape; else $\Omega^{k+1} = \Omega^k$. In this last case, go back to stage (5), using a time step $\tau^{k+1} < \tau^k$.

Remarks 5.

- Of course, the previous description is merely a synthetic, computationally non efficient sketch of the proposed method; depending on the type of objective function J and its derivative, several quantities (such as the solution u_{Ω^k} to the state equation (1)) may be computed only when evaluating $J(\Omega^k)$
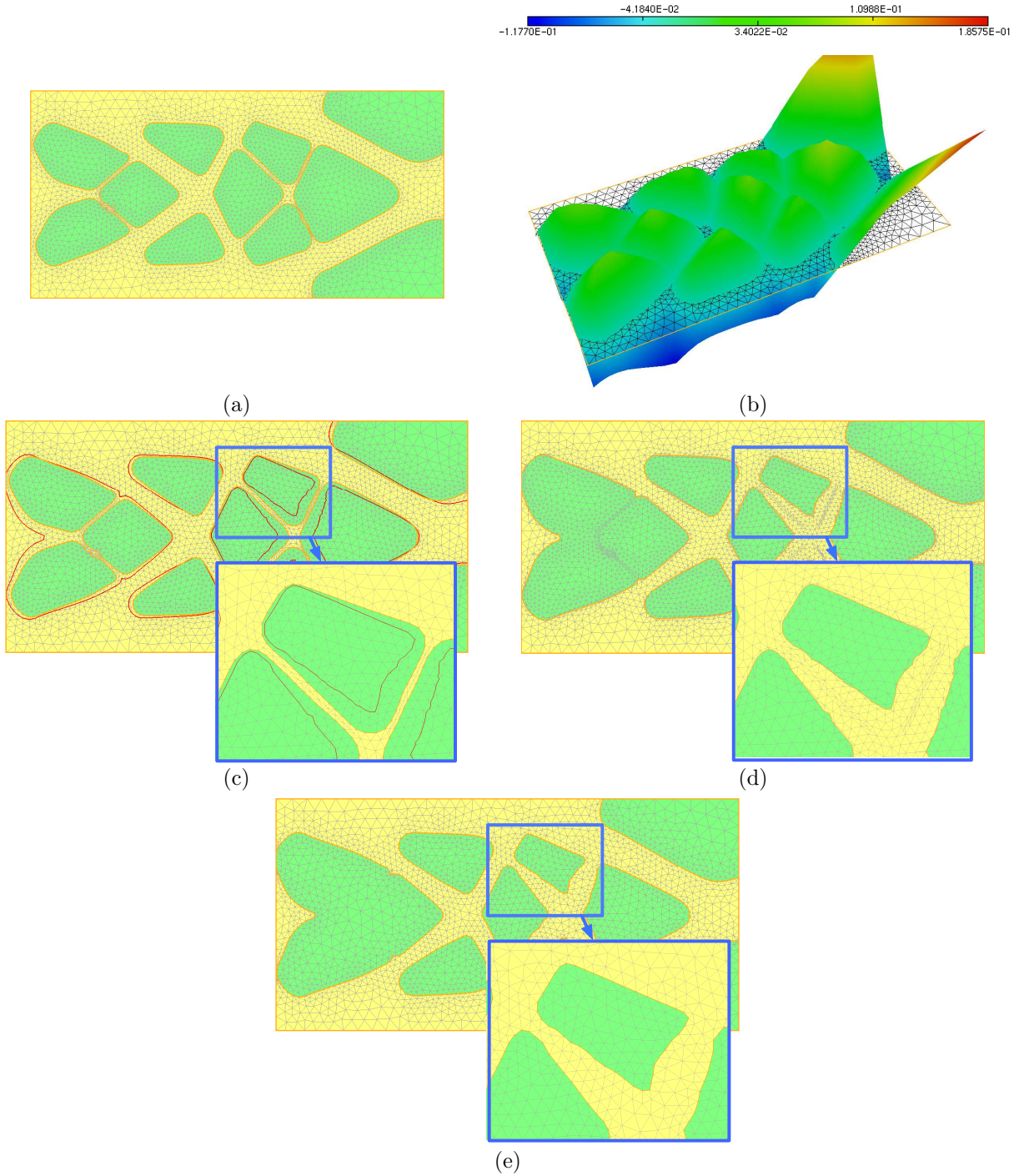
12

FIGURE 5. *(a)* The mesh $\mathcal{T}_{\Omega^k}$ of D accounting for a shape Ω^k; *(b)* the graph of the corresponding level set function ϕ^k; *(c)* advection of ϕ^k according to the velocity field V_{Ω^k} on $\mathcal{T}_{\Omega^k}$; the 0 isoline of the level set function ϕ^{k+1} for the new shape Ω^{k+1} is shown in red and is not yet discretized in the computational mesh; *(d)* explicit discretization of this 0 level set; the obtained mesh $\mathcal{T}_{temp}$ is very ill-shaped; *(e)* high-quality mesh $\mathcal{T}_{\Omega^{k+1}}$ in which the new shape Ω^{k+1} is discretized.

at step (6), then stored for further use when it comes to computing the velocity field V_{Ω^k} during the subsequent step (1).

- This strategy can also be applied to different physical models involving free or moving boundaries [29, 65].
- In numerical practice we did not try to optimize the choice of the descent step. We content ourselves in slightly increasing its value when the objective function is decreasing and, on the contrary, halving it when the objective function is increasing. Note also that it sometimes helps in allowing a small increase of the objective function.

6. Numerical examples

In this section, we present and discuss several numerical test cases, in two and three space dimensions, to assess the interest of the proposed mesh evolution method for shape optimization and illustrate some of its features. All the discussed computations were performed on a laptop computer (MacBook Pro, 2.66 GHz), and, unless stated otherwise, the coefficients of the elastic material at stake are set to $E = 1$ (Young modulus), $\nu = 0.3$ (Poisson ratio).

6.1. **Minimization of the compliance.**

For the sake of simplicity, in all this section, we assume that no body forces are applied, i.e. $f = 0$.

6.1.1. *Two-dimensional examples.*

Our first (simple) examples are concerned with the design of elastic structures with maximal rigidity. The objective function $J(\Omega)$ is thus the *compliance*:

$$(20) \qquad J(\Omega) = \int_\Omega Ae(u_\Omega) : e(u_\Omega)\, dx = \int_{\Gamma_N} g \cdot u_\Omega\, ds,$$

where u_Ω is the solution to (1). This objective function is of the general form (3) with $j = 0$, $k(x, u) = g \cdot u$ on Γ_N, and $k(x, u) = 0$ on Γ. It is well-known that, in this case, the minimization problem for J is self-adjoint, i.e. the adjoint p_Ω involved in the expression (4) of $J'(\Omega)$, solution to (5), is none other than $p_\Omega = -u_\Omega$. So that the problem is not trivial, a volume constraint is imposed under the form of a penalization by a fixed Lagrange multiplier ℓ, as explained in Remark 1.

We start with the benchmark *cantilever* test case: in a working domain D of dimensions 2×1, a beam is clamped around its top and bottom left corners, and surface loads $g = (0, -1)$ are applied on a small area located at the centre of its right-hand side (see the details on Figure 6). The Lagrange multiplier associated to the volume constraint is set to $\ell = 3$, and 200 iterations of the algorithm described in Section 5 are performed. Each mesh $\mathcal{T}_{\Omega^k}$ of D arising in the course of the process has approximately 1500 vertices (and twice as many triangles), and the whole computation takes about 3 minutes. Several intermediate shapes are displayed on Figure 6, and the convergence history for the aggregated objective functional $\mathcal{L}(\Omega) := J(\Omega) + \ell \operatorname{Vol}(\Omega)$ is reported on Figure 8 (left picture). We observe that the shape has been able to change topology without any trouble, while it is exactly meshed at each iteration of the process.

The very same strategy is applied to another benchmark example in structural optimization, namely that of the *optimal mast*: in a T-shaped working domain D, of height 120, width 80 at the top and 40 at the bottom, a mast is clamped around its bottom-left and bottom-right corners, and submitted to surface loads $g = (0, -1)$ around the corners on its arms (see Figure 7). Here, the Lagrange multiplier associated to the volume constraint is set to $\ell = 1$, and 100 iterations of the proposed algorithm are performed. Each intermediate mesh has about 8000 vertices, and the whole computation takes about 5 minutes. Results are shown on Figure 7, and the convergence history for the weighted sum of the compliance and the volume of shapes is reported on Figure 8 (right picture).
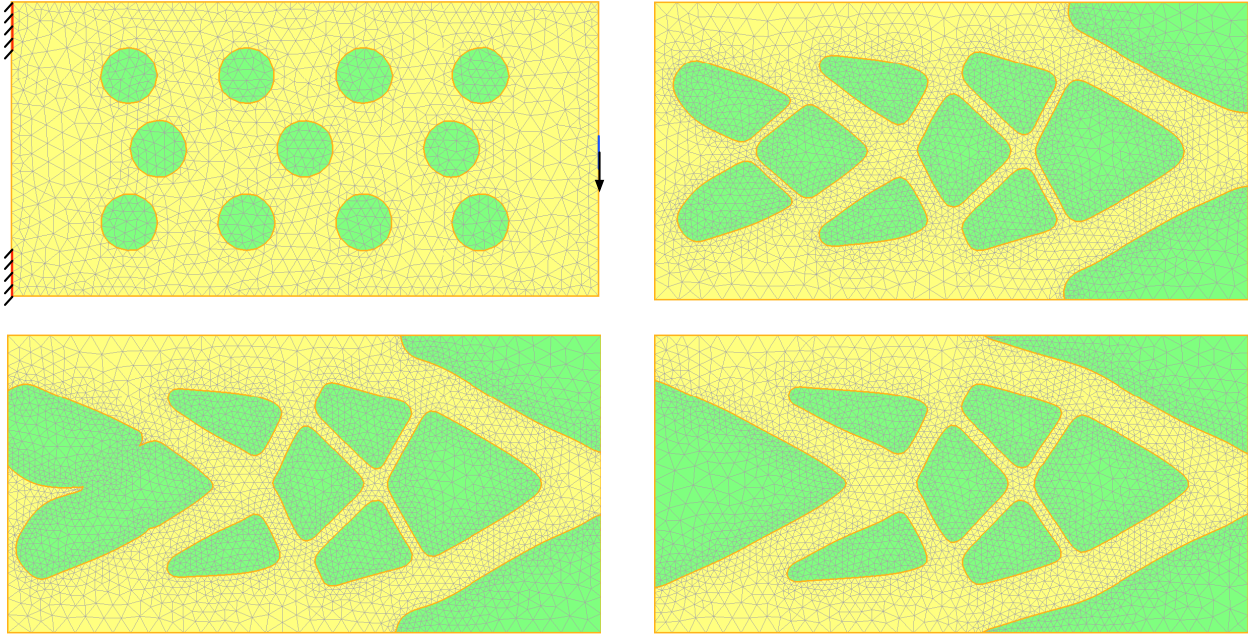
FIGURE 6. *(From top to bottom)* Initial (with boundary conditions), 100th, 114th and final iterations of the $2d$ cantilever test case. The 'inner' domains Ω^k are displayed in yellow, and the 'outer' ones $D \setminus \Omega^k$ in green. Note the ongoing topological change at the 114th iteration.
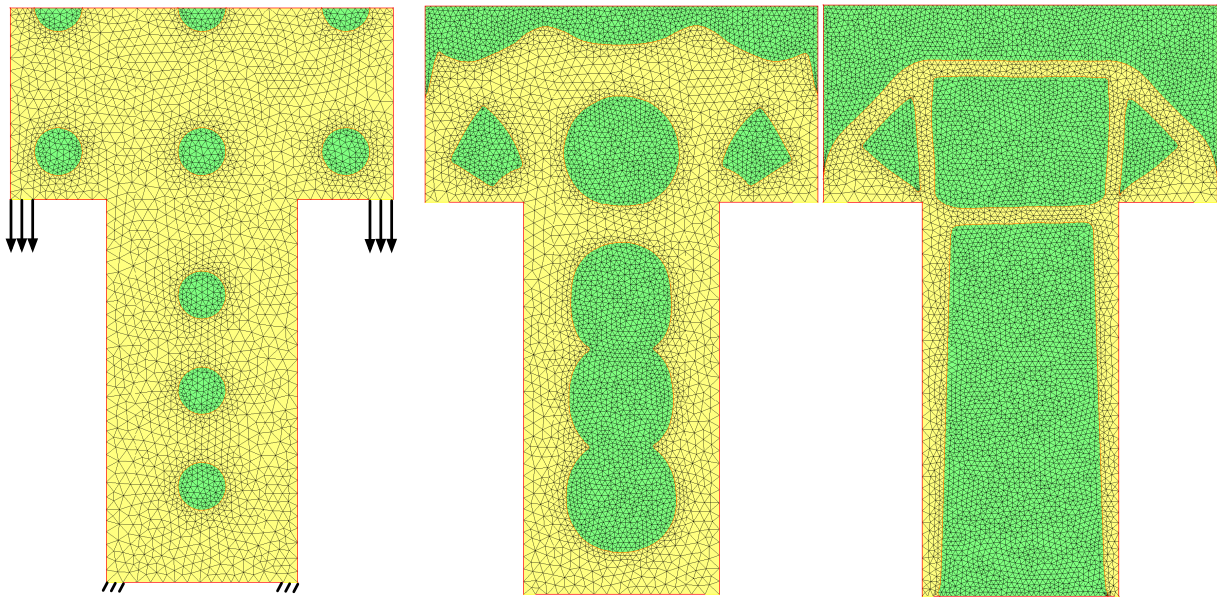

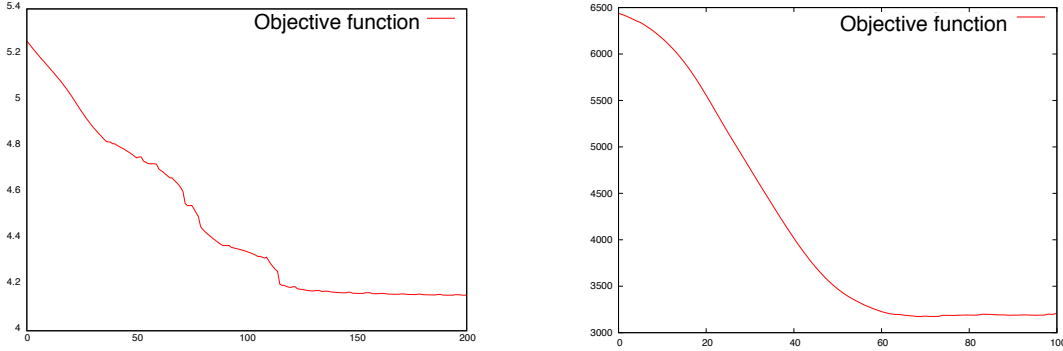FIGURE 7. *(From left to right)* Initial (together with boundary conditions), 30th and final iterations of the optimal mast test case.

FIGURE 8. *(Left)* convergence history for the $2d$ cantilever test case, *(right)* convergence history for the $2d$ mast test case.

6.1.2. *A two-dimensional example using the topological derivative.*

Hadamard's boundary variation method allows to describe the evolution of shapes via deformations of their boundaries. Theoretically speaking, the various shapes obtained in the course of such an evolution process are all diffeomorphic to one another; in particular, they share the same topology. In numerical practice, a small abuse in this setting allows holes to merge (in $2d$), or walls to collide into handles (in $3d$), but holes can never be nucleated in the bulk parts of shapes; this results in a strong dependency of the final design on the topology of the initial one, mostly in $2d$. To circumvent this difficulty, the works [8, 22], based on results of [37, 41, 62], proposed to add an altogether different information to the shape optimization process, namely that of the sensitivity of a shape with respect to the nucleation of a small hole, which we briefly outline now.

Definition 2. *Let Ω be a shape, $x \in \Omega$ a fixed point. For $\rho > 0$ small enough, denote $\Omega_\rho := \Omega \setminus (x + \rho\omega)$, where ω stands for the unit ball in $\mathbb{R}^d$. A real-valued function J of the domain admits a topological derivative $D_T J(\Omega)(x)$ at x if there exists a continuous function $f : \mathbb{R} \to \mathbb{R}$, with $f(0) = 0$ such that the following asymptotic expansion holds in the neighborhood of $\rho = 0$:*

$$J(\Omega_\rho) = J(\Omega) + D_T J(\Omega)(x)\, f(\rho) + \mathrm{o}\left(f(\rho)\right), \ \ with \ \frac{\mathrm{o}(f(\rho))}{f(p)} \xrightarrow{\rho \to 0} 0.$$

The following result, proved in [41], gives the topological derivative of the compliance.

Theorem 2. *The compliance functional $J(\Omega)$ defined by (20) admits a topological derivative at any point $x \in \Omega$, given by the following formula:*

$$D_T J(\Omega)(x) = \frac{\pi(\lambda + 2\mu)}{2\mu(\lambda + \mu)} \left(4\mu Ae(u_\Omega) : e(u_\Omega) + (\lambda - \mu)\mathrm{tr}\left(Ae(u_\Omega)\right)\mathrm{tr}\left(e(u_\Omega)\right)\right)(x)$$

By definition, nucleating an infinitesimally small hole in a shape Ω at a point x where $D_T J(\Omega)(x)$ is negative decreases the value of J. As proposed in [8], we couple the shape optimization method of Section 5 with a periodic use of this topological sensitivity information: every n_{top} iterations, the topological derivative $D_T J(\Omega^k)$ of the actual shape Ω^k is evaluated, and a small percentage (typically, we took the value of 2%) of the elements where it is most negative are removed from Ω^k.

As an example, consider the *optimal bridge* test case, as depicted on Figure 9: a bridge, enclosed in a rectangle D of dimensions 2×1.2 is clamped around its bottom-left and bottom-right corners, and is submitted to surface loads $g = (0, -1)$, applied on a small region around the middle of its bottom side. The Lagrange multiplier for the volume constraint is set to $\ell = 0.1$, and 500 iterations of the aforementioned coupling strategy are performed, with a stage of topological sensitivity analysis replacing the sensitivity analysis using Hadamard's method every $n_{top} = 10$ iterations. Each mesh of D has about 2500 vertices,
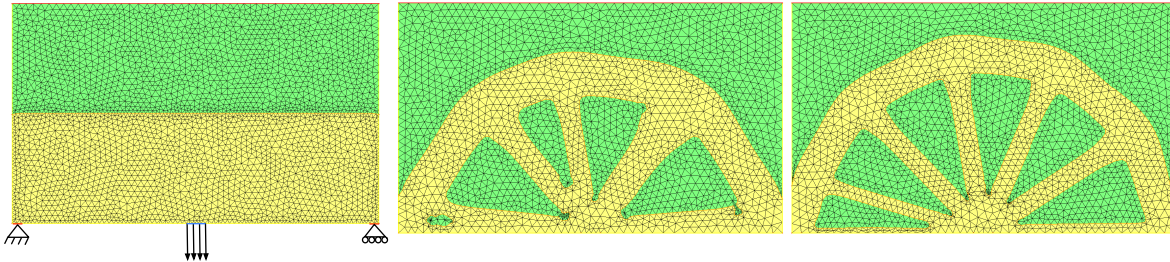
16

FIGURE 9. *(From left to right)* Initial (together with boundary conditions), 60th and final iterations of the $2d$ optimal bridge test case, taking advantage of the information supplied by the topological derivative.
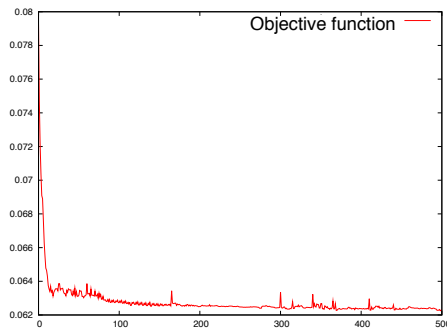

FIGURE 10. Convergence history for the $2d$ optimal bridge test case, using the coupling strategy between shape and topological sensitivity analyses.

and the computation takes less than 10 minutes. Results are displayed on Figure 9, and the convergence history is that of Figure 10. The convergence is not monotone because of the topological gradient steps. Conspicuously, several holes have been nucleated in the course of evolution. Note also that the initial symmetry in the shape has been lost. We shall repeatedly witness this phenomenon in the following (in a less spectacular way however).

6.1.3. $3d$ examples.

We now turn to three-dimensional examples, still minimizing a weighted sum of compliance and volume. Our first example is a cantilever; the computational domain D is a rectangle of dimensions $2.4 \times 5 \times 3$, and the considered shapes are clamped at their right-hand side, while being subject to surface loads, applied on a small area near the centre of their left-hand side (see Figure 11). The Lagrange multiplier for the volume constraint is $\ell = 0.05$ and 80 iterations of the strategy presented in Section 5 are performed. Each mesh has about 16000 vertices (thus approximately six times as many tetrahedra), and the whole computation takes about an hour. Results are displayed on Figure 11, and the associated convergence history is reported to Figure 14 (left picture). Note that some intermediate shapes may show dramatic stretching and that the final shape is nevertheless very regular.

Second, consider the bridge model, depicted in Figure 12: in a working domain D of dimensions $40 \times 200 \times 50$, the considered shapes are clamped on two symmetric parts of their bottom side, and surface loads are applied all over their superior part. The Lagrange multiplier for the volume constraint is $\ell = 100$ and 70 iterations of our algorithm are performed. The average number of vertices of the considered meshes
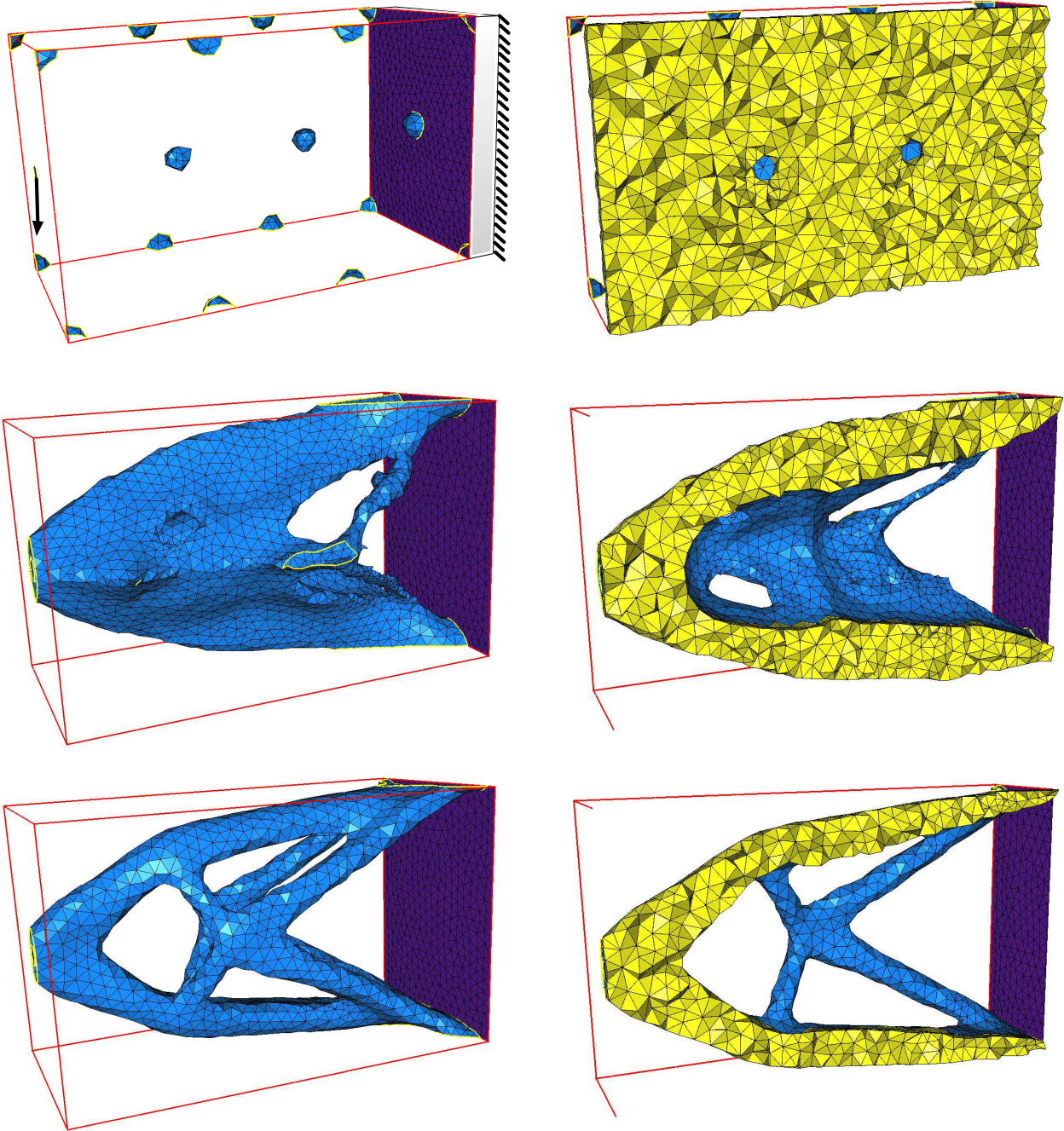
FIGURE 11. *(From top to bottom)* Initial (with boundary conditions), 30th and final (80th) iterations of the $3d$ cantilever test case. Only the boundary $\partial\Omega^k$ of each shape Ω^k is displayed in the left column, and only half of the interior part of each mesh $\mathcal{T}_{\Omega^k}$ is displayed in the right column.

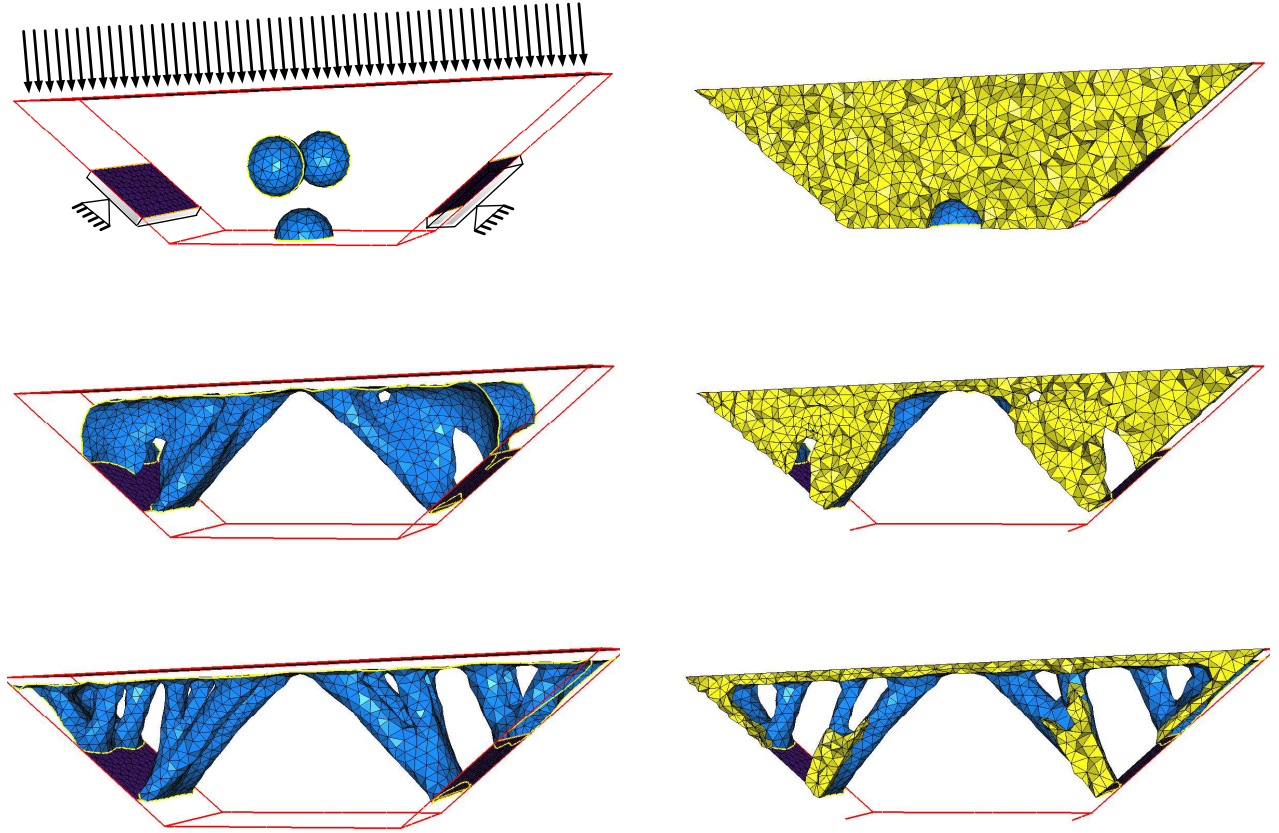FIGURE 12. *(From top to bottom)* Initial (with boundary conditions), 30th and final (70th) iterations of the $3d$ bridge test case.

is 9000, an the computation takes about 45 minutes. See Figure 14 (right picture) for the convergence history.

Figure 13 exemplifies Remark 3, that the remeshing algorithm presented in Section 3.2.2 can be used to produce a high-resolution mesh of the optimal shape: in this particular case, the final shape (or more accurately, the last mesh of D) is enriched into a now one enjoying about 70000 vertices.

6.2. **Multi-loads compliance minimization.**

Still in the context of compliance minimization, we now consider several independent load cases in the optimization process. More specifically, in the general context of Section 2, let $f_i \in L^2(\mathbb{R}^d)^d$, $i = 1, ..., N$ be N body forces, and $g_i \in H^1(\mathbb{R}^d)^d$ be N surface loads, all of them being applied on the same non-optimizable subset Γ_N of the boundaries of shapes in $\mathcal{U}_{ad}$ (of course, each g_i may vanish on a different subset of Γ_N). For any $\Omega \in \mathcal{U}_{ad}$, denote by $u_{\Omega,i} \in H^1(\Omega)^d$ the unique solution to:

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

As in [9], the problem of finding the most rigid shape with respect to the N load cases is expressed as that of minimizing the sum of the individual compliances associated to each load case; the considered objective
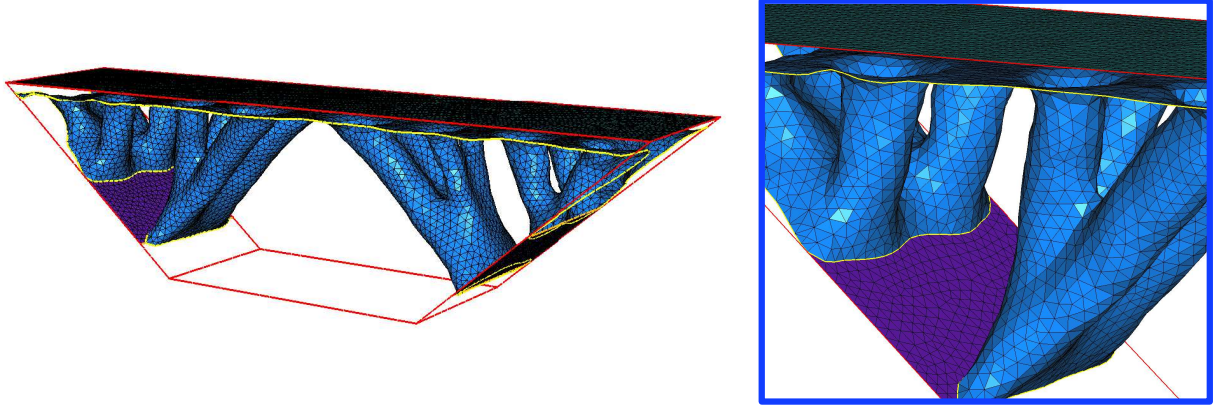
19

FIGURE 13. *(Left)* High-resolution mesh of the final shape obtained in the $3d$ bridge example, *(right)* zoom on the surface mesh.
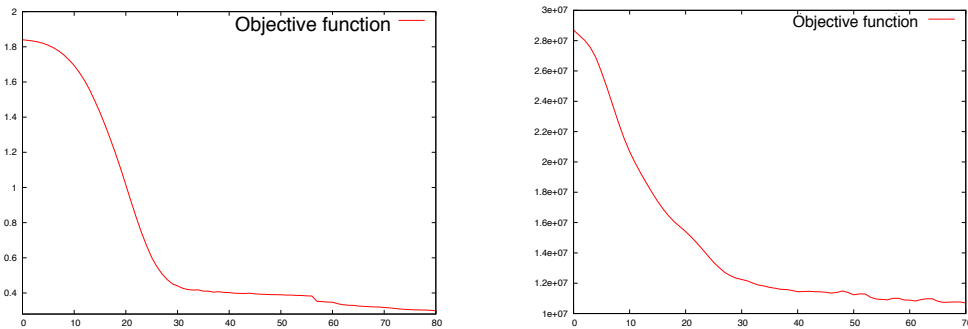

FIGURE 14. *(Left)* convergence history for the $3d$ cantilever test-case, *(right)* convergence history for the $3d$ bridge test-case.

function thus reads:

$$(21) \qquad J(\Omega) = \sum_{i=1}^{N} \int_{\Omega} Ae(u_{\Omega,i}) : e(u_{\Omega,i}) \, dx = \sum_{i=1}^{N} \left(\int_{\Omega} f_i \cdot u_{\Omega,i} \, dx \int_{\Gamma_N} g_i \cdot u_{\Omega,i} \, ds \right).$$

As an example, we consider the *optimal chair* test case, as represented in Figure 15: shapes are embedded in a box of dimensions $0.7 \times 0.5 \times 1$, and submitted to two independent load case: the first one $g_1 = (0, -1)$ is applied on the seat of the chair, and the second one $g_2 = (-1, 0)$ is applied on the back (in both cases, no body forces are applied). Function (21) is minimized after a volume constraint has been incorporated under the form of a fixed Lagrange multiplier $\ell = 200$: 100 iterations of our algorithm are performed, for a total computational time of approximately 90 minutes (each mesh enjoying about 11000 vertices). Results are displayed on Figure 15 (see also Figure 16 for the convergence history).

6.3. **Chaining topological and geometric optimization.**

In this section, we elaborate on the first point of Remark 3 about the possibility and benefits of combining the mesh evolution method for shape optimization of this article with the 'classical' level-set based structural optimization method on a fixed mesh, as in [11, 68]. We do not claim that they *must* be chained but it can turn out to be interesting for the following various (and debatable) reasons. First, in an industrial context, an engineer or a researcher may have his favorite software for optimizing a structure and use it as
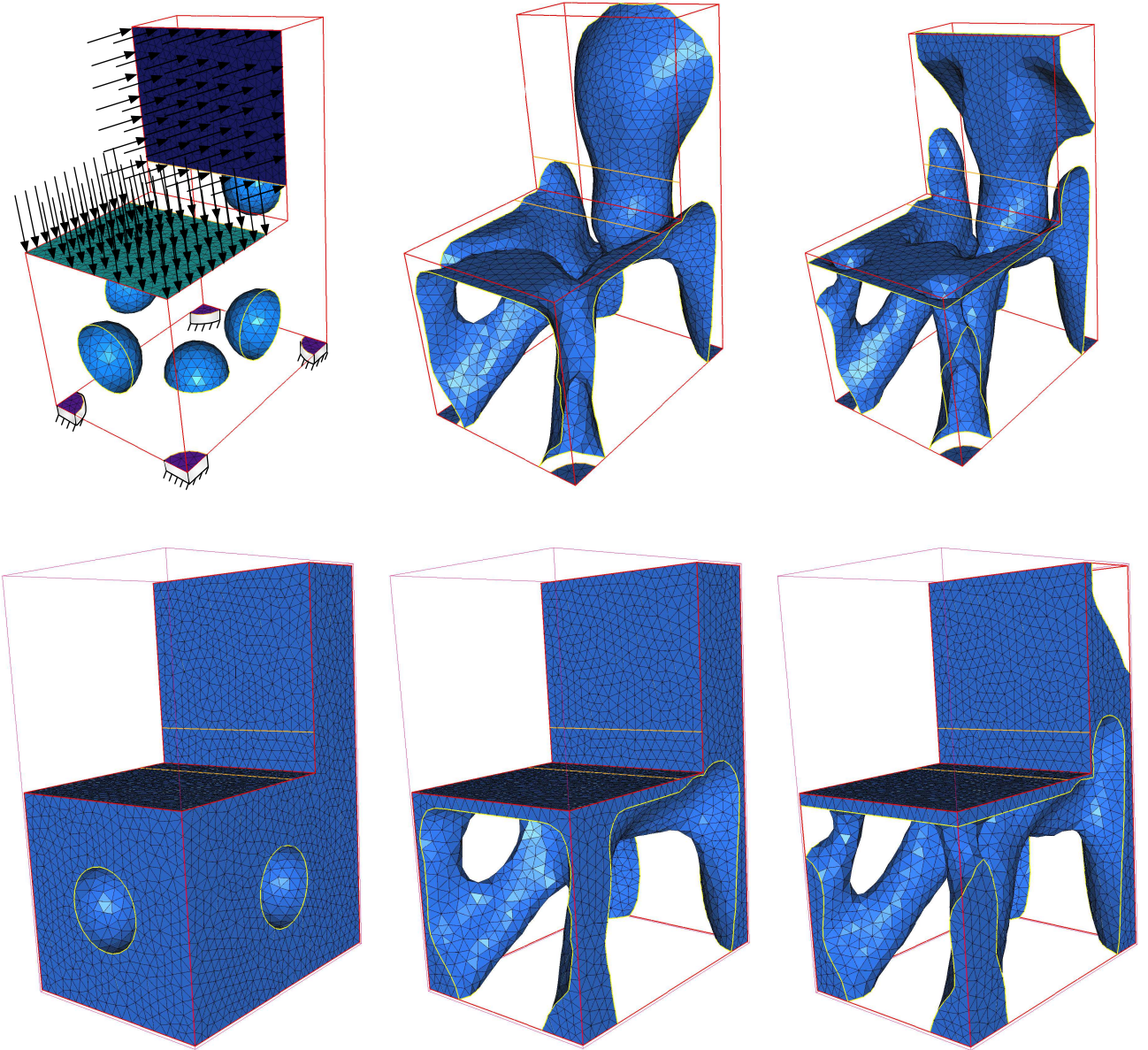
FIGURE 15. *(From left to right)* Initial (with boundary conditions), 50th and final (100th) iterations of the $3d$ optimal chair test case. To help visualization, the whole boundaries of shapes (and not only that corresponding to the 0 level set of the evolving implicit function) are displayed on the lower row.

a pre-processor (delivering an initial guess) to our mesh evolution method. In other words, the idea is to see how to couple two methods and how much the optimal result can be improved. Second, the chaining of these two methods can benefit from their different properties and assets:

- The classical level-set optimization method may prove faster in terms of CPU time, for at least two reasons: first, it does not involve remeshing; second, it can generally be carried out on a structured mesh of the computational domain D, which allows for the use of unexpensive finite difference schemes for solving the level set advection equation (13) or the Hamilton-Jacobi equation (14) accounting for the evolution of the shape. Note however that this argument is not fully obvious
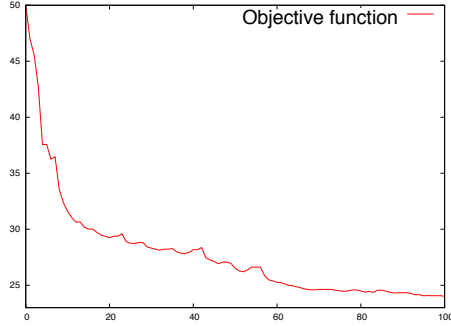
FIGURE 16. *Convergence history for the optimal chair test case.*

since the proposed mesh evolution method is performing FEM analysis only on a smaller adapted submesh of the whole mesh of D (which involves fewer elements than in the case of the classical level set method).

- The classical level-set optimization method has features which may make it preferable for a first try. For example, using Cartesian meshes may guarantee symmetric designs. Or it may be felt as being more robust because people have more experience with it.
- On the other hand, the proposed mesh evolution method definitely yields a better evaluation of the mechanical properties since the shapes are exactly meshed. This allows for a more precise calculation of the shape gradient, a desirable feature in the final steps of an optimization algorithm where subtle changes affect the shape. This expected enhanced accuracy of mechanical computations is especially crucial for computing bounds on the Von Mises stress.

Let us now get into specifics; we examine the following two-stage strategy for minimizing a function $J(\Omega)$:

(1) *Optimization of the shape using the 'classical' level set method:* the working domain D is endowed with a *fixed* mesh $\mathcal{T}$ (which may be simplicial, Cartesian, etc...), and shapes $\Omega \subset D$ are consistently described by a corresponding level set function ϕ (discretized on $\mathcal{T}$). The main source of approximation of this class of shape gradient algorithms (which is also its fundamental difference with the method of this paper) is that no mesh of a shape $\Omega \subset D$ is available when it comes to performing the necessary computation of the solution u_Ω to (1) (or the adjoint state p_Ω) for the derivation of a descent direction for J. Hence, the *Ersatz material approach* is used, whereby the void part $D \setminus \Omega$ is filled with a 'very soft' material of Hooke's law εA, $\varepsilon \ll 1$, so that the problem (1) is approximated by the following one, posed on D:

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(A_\Omega e(u)) & = & f & \text{in D} \\ u & = & 0 & \text{on } \Gamma_D \\ A_\Omega e(u)n & = & g & \text{on } \Gamma_N \end{array} \right. ,$$

where the total Hooke's tensor A_Ω is defined as:

$$\forall x \in D, \ A_\Omega(x) = \left\{ \begin{array}{ll} A & \text{if } x \in \Omega \\ \varepsilon A & \text{if } x \in D \setminus \Omega \end{array} \right. .$$

This step ends with a temporary 'optimal' shape $\widetilde{\Omega}$, known as a level set function $\widetilde{\phi}$, defined on mesh $\mathcal{T}$.

(2) *Optimization of the shape using the mesh evolution method:* The resulting shape $\widetilde{\Omega}$ from the first step is explicitly discretized in the computational mesh $\mathcal{T}$, which produces a new mesh $\mathcal{T}_{\widetilde{\Omega}}$ of D in which $\widetilde{\Omega}$ is enclosed as a submesh. From this point, the algorithm of Section 5 is applied, retaining exactly the same parameters (loads, Lagrange multipliers, etc...) as in the first stage (except of course for the Ersatz coefficient ε which no longer serves any purpose), and produces a new 'optimal' shape, say Ω^*.
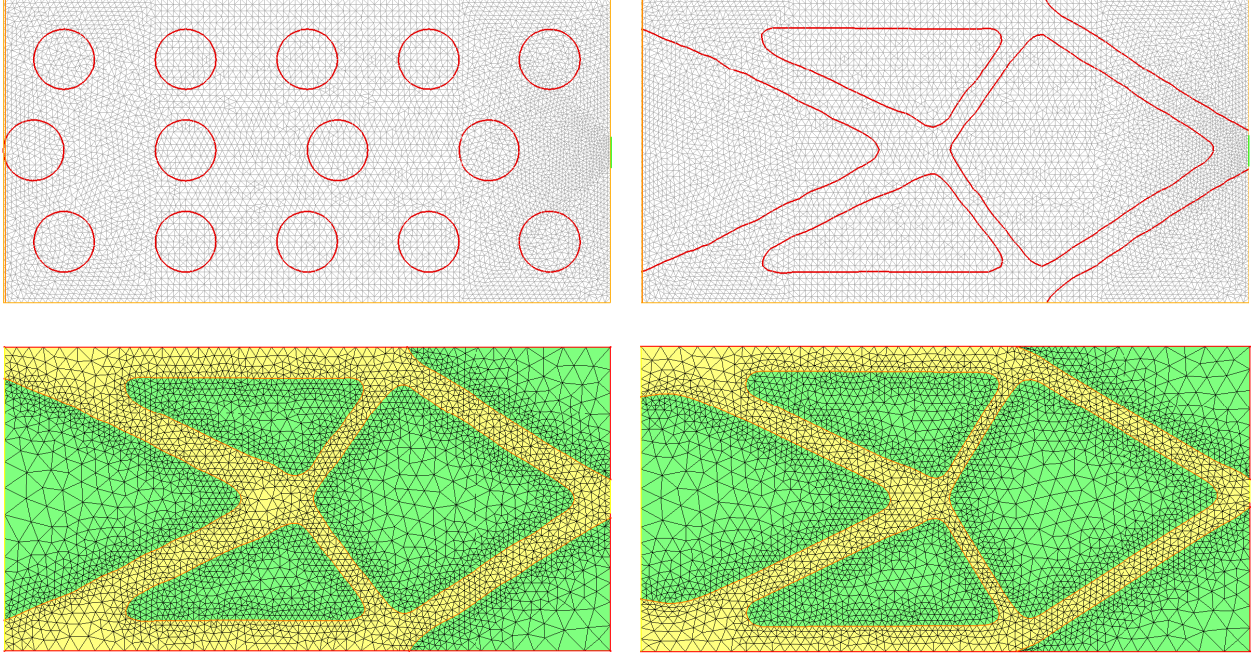
FIGURE 17. *(Top)* Initial and final iterations of the 2*d* Cantilever test case, using the level set method for shape optimization on a fixed unstructured mesh. The 0 level set accounting for the shape of interest is displayed in red. *(Bottom-left)* The 0 level set of obtained during the first stage is discretized into the computational mesh; *(Bottom-right)* final result of the combination of both methods.

To appraise this procedure, we limit ourselves to the case of the aggregated sum $\mathcal{L}(\Omega)$ of the compliance (20) and the volume Vol(Ω) as an objective function of the domain, and first consider the two-dimensional cantilever example of Section 6.1.1, using the same parameters as those introduced then.

The first stage is performed on a fixed triangular mesh $\mathcal{T}$ of the working domain D containing 6518 vertices. The coefficient for the weak material is $\varepsilon = 1.e^{-3}$, and 200 iterations of the fixed mesh level set method are performed using `FreeFem++` [46], [58] to produce the intermediate 'optimal' shape $\widetilde{\Omega}$.

As for the second stage, we apply the algorithm of Section 5 for another 200 iterations, using the exact same parameters as in the first stage. All the meshes of this second sequence have more or less 3500 vertices. Results are displayed on Figure 17 (see Figure 18 for the convergence histories). Note that the respective shapes $\widetilde{\Omega}$ and Ω^* obtained at the end of stage (1) and (2) are qualitatively different, and that the final Ω^* is a noticeable improvement of the first 'optimal shape' $\widetilde{\Omega}$. Note also the non negligible gap between the values of $\mathcal{L}(\widetilde{\Omega})$ depending on whether it is computed by using the Ersatz material approximation or not (see Table 1).

The same strategy is applied to the 3*d* cantilever test case. Here, the setting of the problem is slightly different from that of Section 6.1.3: the working domain D is now a $2 \times 1 \times 1$ box; shapes are clamped at their left-hand side and a point load is applied at the centre of the right-hand side.

During stage (1), D is equipped with a Cartesian mesh of size $40 \times 20 \times 20$ (18081 vertices), and the resulting optimal shapes are courtesy of G. Michailidis [50]. Two examples are presented, associated to different Lagrange multipliers $\ell = 100$ and $\ell = 200$ for the volume constraint. Results are displayed on Figure 19. In both cases, stage (2) converges within only ten iterations (hence, only the final values of the objective functions are recorded in Table 1), and the intermediate 'optimal' shapes $\widetilde{\Omega}$ are significantly improved (even topological changes occurred !) by the respective final 'optimal' shapes Ω^*. In the case $\ell = 100$ (resp. $\ell = 200$), the average number of vertices of the meshes arising during stage (2) is 15000 (resp. 12000).
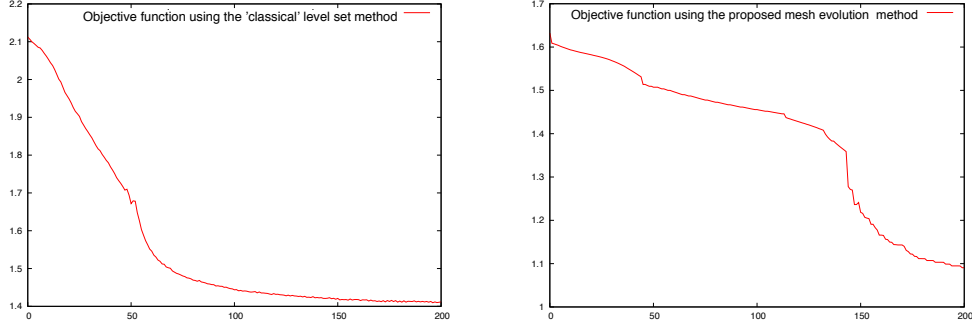
23

FIGURE 18. Convergence histories for the chaining procedure of topological and geometric optimization methods; *(left)* the 'classical' level set method is performed on a fixed triangular mesh for 200 iterations; *(right)* the resulting shape from the first step is used as an input for the mesh evolution method, which is carried out for another 200 iterations.

	$\mathcal{L}(\widetilde{\Omega})$ with the Ersatz material approximation	$\mathcal{L}(\widetilde{\Omega})$ without the Ersatz material approximation	$\mathcal{L}(\Omega^*)$
$2d$ Cantilever	1.41182	1.632306	1.090604
$3d$ Cantilever, $\ell = 100$	188.5576	191.876896	162.661392
$3d$ Cantilever, $\ell = 200$	259.2133	258.723087	221.106033

TABLE 1. Values of the objective functions at different stages of the chaining strategy of the 'classical' level set method and the mesh evolution method for structural optimization. Note the significant discrepancy between the values of $\mathcal{L}(\widetilde{\Omega})$ when appraised with, or without using the Ersatz material approximation in the $2d$ case (first line).

6.4. **Multi-materials compliance minimization.**

In this section, we discuss a model which generalizes the framework of Section 2, namely that of *multi-phase* shape optimization. For simplicity we present the case of two-phases and we refer to Remark 6 for the extension to multiple phases using the color level set method of [66]. Let D be a fixed working domain containing two materials, referred to as 0 and 1, with different properties reflected by their respective Hooke's tensor A_0 and A_1. They occupy two smooth subdomains Ω^0 and Ω^1, with $\Omega^1 = D \setminus \overline{\Omega^0}$, and for the sake of simplicity we assume that Ω^0 does not touch the boundary ∂D, i.e. $\partial\Omega^0 \cap \partial D = \emptyset$. The domain D is clamped on a region Γ_D of its boundary ∂D, and surface loads $g \in H^1(\mathbb{R}^d)^d$ are applied on another subset $\Gamma_N \subset \partial D$, disjoint from Γ_D. Body forces $f \in L^2(\mathbb{R}^d)^d$ are also applied and the induced displacement $u_{\Omega^0} \in H^1(D)^d$ is the unique solution to:

$$
(22) \qquad \left\{ \begin{array}{rcll} -\mathrm{div}\left(A_{\Omega^0} e(u)\right) & = & f & \text{in D} \\ u & = & 0 & \text{on } \Gamma_D \\ A_{\Omega^0} e(u) n & = & g & \text{on } \Gamma_N \end{array} \right. ,
$$

where the total Hooke's tensor A_{Ω^0} is defined over D as:

$$
\forall x \in D, \;\; A_{\Omega^0}(x) = \left\{ \begin{array}{ll} A_0 & \text{if } x \in \Omega^0 \\ A_1 & \text{if } x \in \Omega^1 \end{array} \right. .
$$

We consider the compliance $J(\Omega^0)$ of the structure as an objective function of the subdomain Ω^0:

$$
(23) \qquad J(\Omega^0) = \int_D A_{\Omega^0} e(u_{\Omega^0}) : e(u_{\Omega^0}) \, dx = \int_D f \cdot u_{\Omega^0} \, dx + \int_{\Gamma_N} g \cdot u_{\Omega^0} \, ds.
$$

In truth, the new optimization variable is the interface Γ between the two phases, defined by $\Gamma = \partial\Omega^0 \cap \partial\Omega^1$. The shape derivative of (23) was computed in [12].
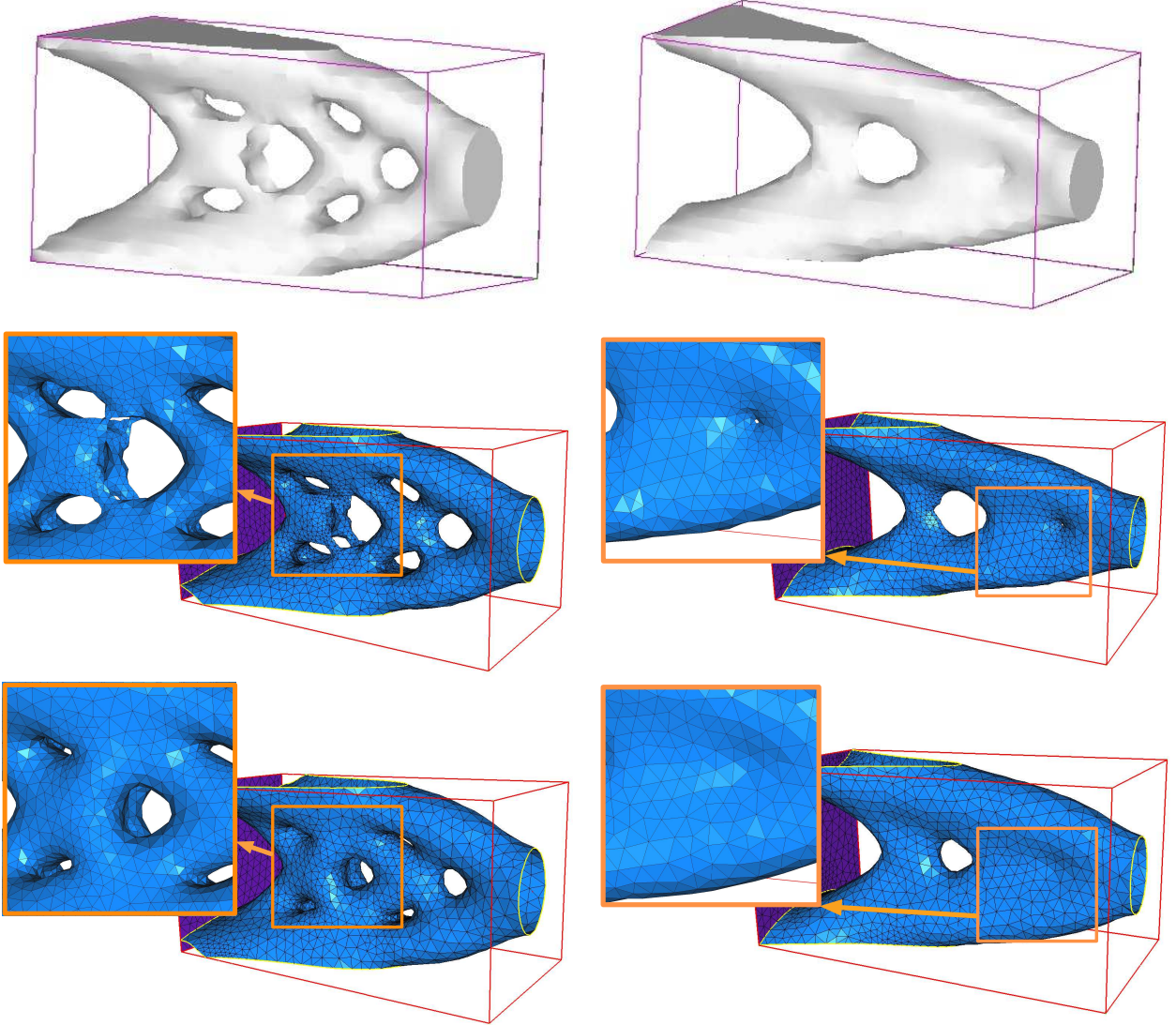
24

FIGURE 19. *(From top to bottom):* 0 level set of the implicit function for $\widetilde{\Omega}$, associated discretization in the mesh of D, and final shape Ω^*, *(left column)* using a Lagrange multiplier $\ell = 100$, *(right column)* using a Lagrange multiplier $\ell = 200$.

Theorem 3. *The shape derivative of the compliance J defined by (23) reads*

$$J'(\Omega^0)(\theta) = \int_\Gamma \mathcal{D}(u_{\Omega^0}, u_{\Omega^0})\, \theta \cdot n\, ds,$$

(24) $$\mathcal{D}(u, u) = -\sigma(u)_{nn}[e(u)_{nn}] - 2\sigma(u)_{n\tau} \cdot [e(u)_{n\tau}] + [\sigma(u)_{\tau\tau}] : e(u)_{\tau\tau}.$$

where $\mathcal{M}_{nn}$, $\mathcal{M}_{n\tau}$ and $\mathcal{M}_{\tau\tau}$ are the minors of a tensor field $\mathcal{M} = \begin{pmatrix} \mathcal{M}_{\tau\tau} & \mathcal{M}_{\tau n} \\ \mathcal{M}_{n\tau} & \mathcal{M}_{nn} \end{pmatrix}$ expressed in an orthonormal basis of $\mathbb{R}^d$ obtained by assembling an orthonormal basis of tangent vectors τ to Γ with its normal vector n, $[\cdot] = \cdot^1 - \cdot^0$ denotes the jump through Γ, and $\sigma(u) = A_{\Omega^0}\, e(u)$.

As explained in [5], this problem is very difficult to handle in a fixed mesh framework where the interface Γ is not exactly meshed (at least without any change in the formulation). The difficulty is that formula (24) brings into play the transmission conditions at the interface Γ, which cannot be accurately approximated

25

using Lagrange finite element methods for solving (22) since the interface is merely captured but not tracked. On the contrary, when Γ is explicitly discretized in a mesh of D, then the jump of the derivatives $[e(u)_{nn}]$, $[e(u)_{n\tau}]$, $[\sigma(u)_{\tau\tau}]$ can be accurately computed by standard Lagrange finite element methods. Therefore, it is a unique feature of our proposed method to be able to handle this type of multiphase geometric optimization problem without any additional ingredients.

As a test case, we consider a $3d$ box of dimensions $40 \times 200 \times 60$ as D, which is clamped around its four bottom corners, and submitted to surface loads on a region near the center of its upper side. It is filled with two materials: material 0 with Young modulus $E_0 = 1$ and a weaker material 1 with $E_1 = 0.3$ (both have the same Poisson ratio $\nu_1 = \nu_0 = 0.3$). The compliance (23) of the total structure D is minimized, and a constraint over the volume $\mathrm{Vol}(\Omega^0)$ of the stronger phase is imposed by means of a fixed Lagrange multiplier $\ell = 0.02$. After 100 iterations of our algorithm the results are displayed on Figure 20 (see also Figure 21 for the convergence history). As expected, the stronger material connects the regions where the shape is clamped to the one where loads are applied. A portion of the stronger material at the bottom of the beam allows it to withstand bending.

Remark 6. As examplified in [5], this model extends to the optimization of the interfaces between more than two materials: one then relies on the use of m level set functions for representing up to 2^m materials, according to the so-called *color level set method* [66].

6.5. **Application to worst-case design in shape optimization.**

The goal of this subsection is to apply our method to a worst-case design optimization. We choose the framework introduced in [4] to deal with the shape optimization of the worst-case compliance when 'small' perturbations are expected on the applied body forces to the structures.

Let us briefly outline the main ideas of this setting. Still in the context of Section 2, we now foresee that unknown perturbations of 'small' amplitude m may alter the body force term f, which then becomes $(f + \chi\xi e)$, where:

- $\chi \in L^\infty(\mathbb{R}^d)$ is the (known) characteristic function of the zone where perturbations are expected,
- $\xi \in L^2(\mathbb{R}^d)$ is the amplitude of perturbations, which we only know to satisfy $||\xi||_{L^2(\mathbb{R}^d)} \leq m$,
- $\widehat{e} \in \mathbb{R}^d$ is a unit vector (fixed for simplicity), indicating the direction of the expected perturbations.

For any perturbation term $\xi \in L^2(\mathbb{R}^d)$ with $||\xi||_{L^2(\mathbb{R}^d)} \leq m$, denote as $u_{\Omega, f+\chi\xi\widehat{e}}$ the solution to the perturbed linear elasticity system:

$$
\begin{cases}
-\mathrm{div}(Ae(u)) & = & f + \chi\xi\widehat{e} & \text{in } \Omega \\
u & = & 0 & \text{on } \Gamma_D \\
Ae(u)n & = & g & \text{on } \Gamma_N \\
Ae(u)n & = & 0 & \text{on } \Gamma
\end{cases}.
$$

We would like to minimize the worst (maximal) possible compliance over the set of all admissible perturbations. Thus we introduce the 'worst-case' functional $\mathcal{J}(\Omega)$, defined by

$$
(25) \qquad \mathcal{J}(\Omega) = \sup_{\substack{\xi \in L^2(\mathbb{R}^d) \\ ||\xi||_{L^2(\mathbb{R}^d)} \leq m}} \left(\int_\Omega (f + \chi\xi\widehat{e}) \cdot u_{\Omega, f+\chi\xi\widehat{e}} \, dx + \int_{\Gamma_N} g \cdot u_{\Omega, f+\chi\xi\widehat{e}} \, ds \right).
$$

If the perturbation amplitude m is small, a fair approximation of $\mathcal{J}(\Omega)$ is obtained by linearizing the compliance

$$
\xi \mapsto \left(\int_\Omega (f + \chi\xi\widehat{e}) \cdot u_{\Omega, f+\chi\xi\widehat{e}} \, dx + \int_{\Gamma_N} g \cdot u_{\Omega, f+\chi\xi\widehat{e}} \, ds \right),
$$

then taking the supremum of the linearized quantity over $||\xi||_{L^2(\mathbb{R}^d)} \leq m$, which can be computed explicitly. This leads (after some computations) to considering the following *approximate worst-case functional*, hereafter denotes as $J(\Omega)$, defined by:

$$
J(\Omega) = \int_\Omega f \cdot u_\Omega \, dx + \int_{\Gamma_N} g \cdot u_\Omega \, ds + 2m||\chi u_\Omega \cdot \widehat{e}||_{L^2(\mathbb{R}^d)},
$$

where $u_\Omega := u_{\Omega, f}$ stands for the solution to the unperturbed system. This formula is easily put under the general form (3).
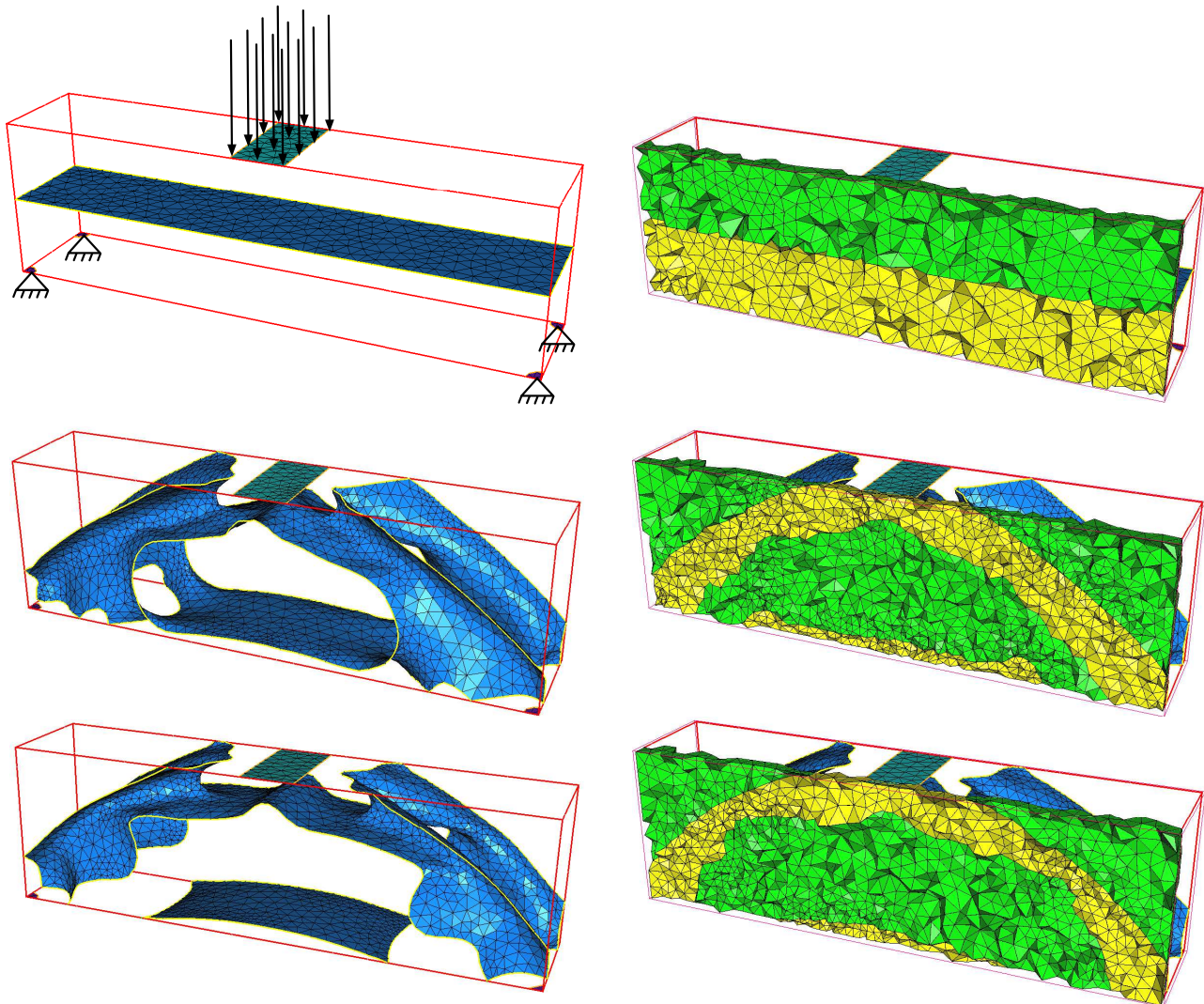
FIGURE 20. *(From top to bottom)* Initial (with boundary conditions), 50th and final (100th) iterations of the multiphase beam test case. The stronger material is the one displayed in yellow in the right-hand column.
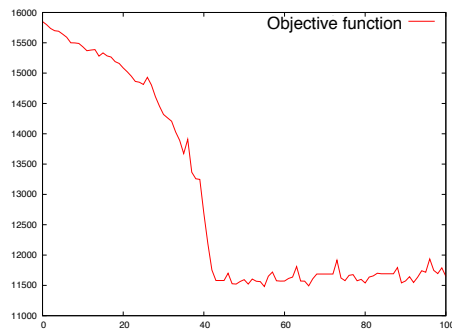

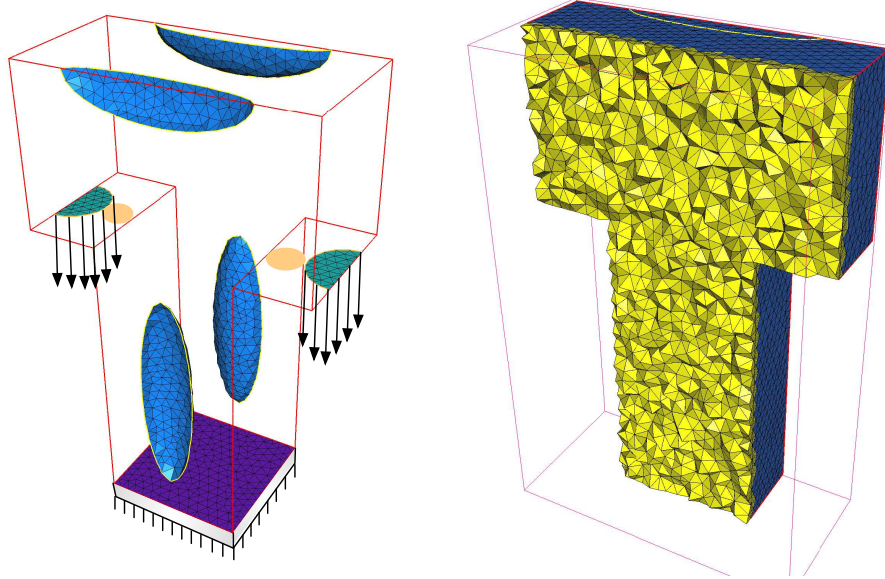FIGURE 21. Convergence history for the multi-material 3*d* Beam test case.

FIGURE 22. *(Left)* initial shape in the perturbed optimal mast test-case, together with boundary conditions for the test-case; *(right)* a cut in the corresponding 3*d* mesh.

Remark 7. Note that this problem of minimizing the worst-case compliance (25) has actually already been addressed in [24, 44] by a completely different method, which allows to work directly with the exact worst-case function $\mathcal{J}(\Omega)$ given by (25) (and is thus certainly more accurate than the presented approximation). However, the above process of linearized worst-case design is easily extended to more general objective functions [4].

As an illustration, consider the model of Figure 22, which is a variation of the well-known *optimal mast* test case: a mast, enclosed in a T-shaped box of dimensions $40 \times 80 \times 126$ is clamped on its bottom side, and submitted to surface loads $g = (0, 0, -1)$ on two areas at the extremities of its arms. Body forces in the unperturbed state are set to $f = 0$, and vertical perturbations (i.e. $\widehat{e} = (0, 0, -1)$) are expected, which are localized on the two yellow regions on the arms. We perform 100 iterations of the proposed algorithm for three different values of m, namely $m = 0, 5, 10$, using, for the sake of simplicity, the same Lagrange multiplier $\ell = 5$ in the three cases, which is then associated to different volume constraints. Each mesh produced in the course of the process is approximately worth 12000 vertices, and the total computational time is about 90 minutes for $m = 5, 10$, and less than an hour for $m = 0$ (since no adjoint state is involved then). Results are reported in Figure 23, and convergence histories lie in Figure 24.

6.6. Stress criterion minimization.

Our last example is concerned with the design of structures minimizing their stress. The objective function at stake is then:

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

where $k \in L^\infty(\mathbb{R}^d)$ is a localization factor, $\sigma(u) := Ae(u)$ is the stress tensor associated to a displacement u and $|.|$ stands for the Frobenius matrix norm. Strictly speaking, this objective function is not of the form (3), since it depends on the derivatives of the displacement u_Ω rather than on the displacement itself. However, the computation of its shape derivative proves very similar to that of (3) and has been performed e.g. in [10]: it notably features an adjoint state p_Ω, which differs from that involved in the test cases of section 6.5 (see[10] for details).
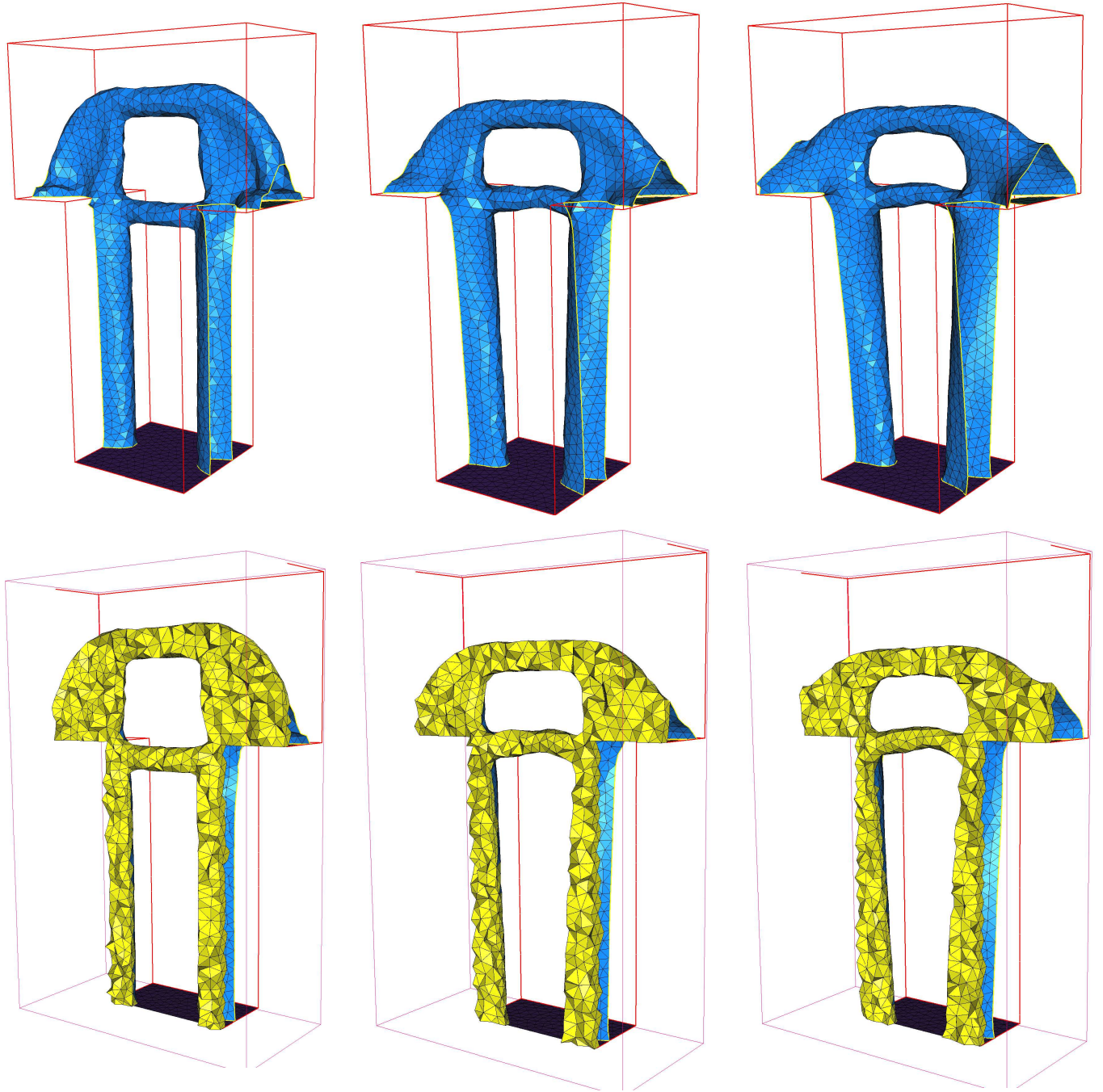
FIGURE 23. *(Upper range)* optimal shapes in the perturbed optimal mast test case for perturbations of amplitude $m = 0, 5, 10$; *(lower range)* cuts in the corresponding $3d$ meshes.

As an example, let us consider the $3d$ *L-Beam* test case, as depicted on Figure 25: shapes enclosed in a L-shaped box of dimensions $2 \times 1 \times 2$ are clamped on their upper side, and submitted to a pointwise unit load, applied at the centre of their front side (once again, no body forces are applied). The localization factor $k(x)$ is chosen to be 1 everywhere, except on a small region around the application point of the load; it is thus expected that the considered objective function does not emphasize the importance of the stress singularity arising in the region. A volume constraint is enforced by means of a fixed Lagrange multiplier $\ell = 200$, and
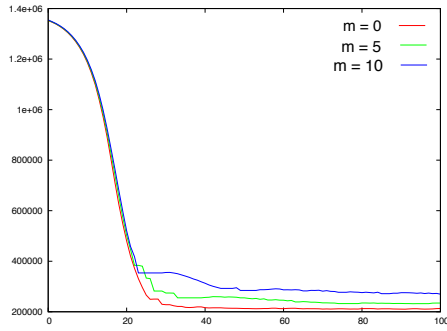
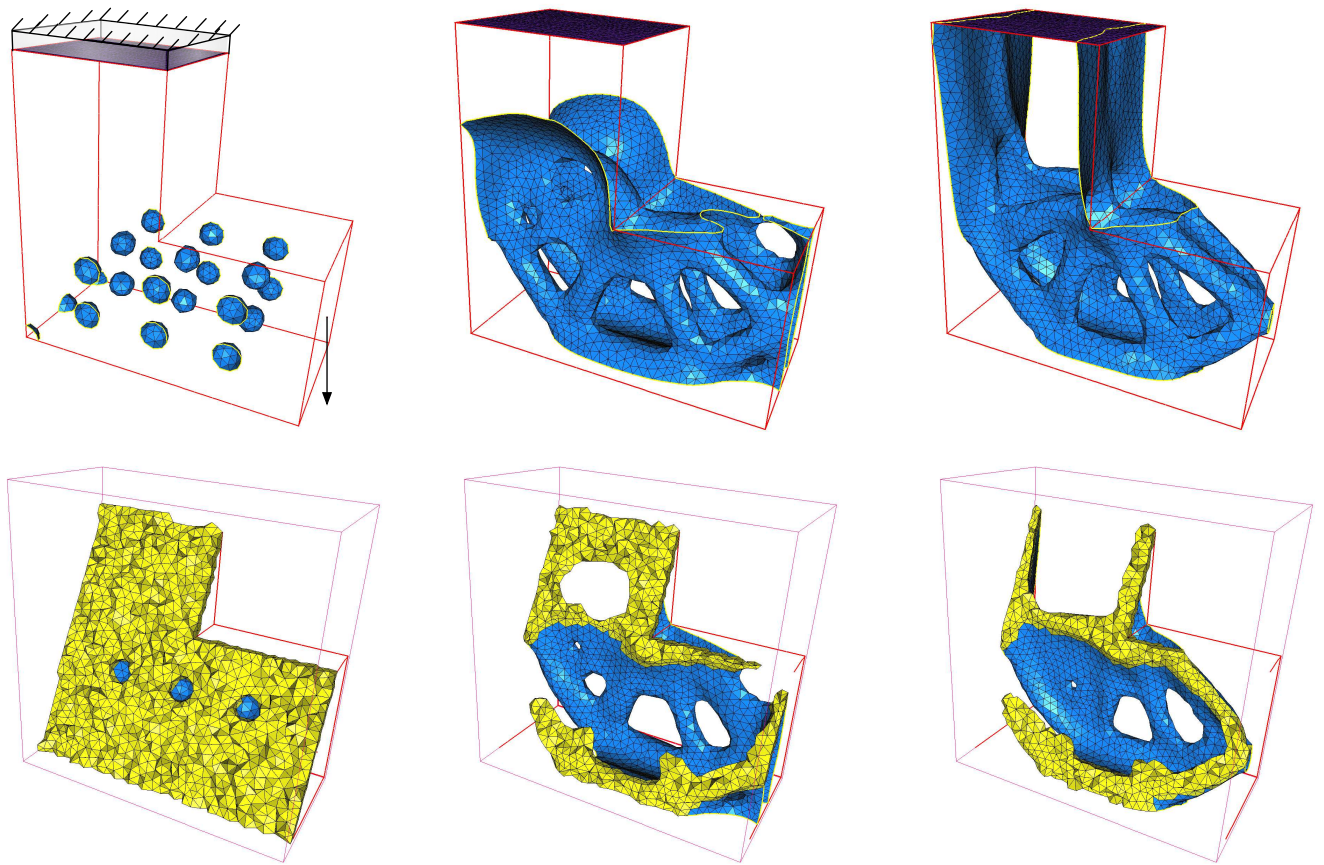FIGURE 24. *Convergence histories for the robust mast test case.*

FIGURE 25. *(From top to bottom)* Initial (with boundary conditions), 50th and final iterations of the $3d$ L-Beam test case.

100 iterations of our algorithm are performed. Each mesh arising in the optimization sequence has about 17000 vertices, and the whole computation takes about 100 minutes. Results are depicted in Figure 25 (see Figure 26 for the convergence history).

7. CONCLUSIONS AND PERSPECTIVES

In light of our numerical experience we make the following general comments on our method for dealing with mesh evolution in shape optimization.
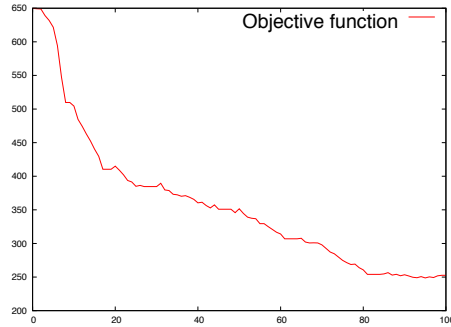
FIGURE 26. *Convergence history for the 3d L-Beam test-case.*

- The proposed method is able to handle dramatic changes in shapes (even topological changes), while keeping an explicit mesh of them at each iteration of the evolution process. Some intermediate shapes may show very stretched features (especially when a topological change occurs), which ineluctably are meshed with stretched elements. Surprisingly, it does not cause any trouble for solving the linearized elasticity system although it might be more delicate for other mechanical models like geometrically nonlinear elasticity.
- By definition the mesh is changing at each iteration. Therefore, any comparison of two sucessive evaluations of the objective function is prone to systematic approximation errors. In some sense, it makes our method slightly more sensitive to numerical errors than the 'classical' level set method, working on a fixed mesh. This may explain the rough behavior of some of the convergence histories in our numerical test cases. Nevertheless, it did not cause any serious trouble in the final convergence.
- It also features a greater accuracy. This is especially crucial in the case of objective functions where an explicit discretization of the boundaries of shapes is helpful (e.g. when it comes to computing the stress developed in shapes. This feature is also especially patent in the example of Section 6.4 where the transmission conditions between two phases can be properly discretized since the interface is explicitly meshed. Even for extremely simple benchmark, like the 2d minimal compliance cantilever, Section 6.3 demonstrated that the mesh evolution method manages to improve 'optimal' results produced by the fixed mesh level set method.
- The proposed method does not allow to retain the symmetry of shapes, as can be observed on several of the above examples (however, it sometimes happens that the evolution is non symmetric, and the final result miraculously is !). It seems very difficult to enforce any symmetry in shapes with our fully unstructured method unless all the computations are held only on a representative subdomain, which is replicated by symmetry to get the corresponding shape.

Eventually, we believe that, among others, the following topics could be worth some further investigations in future works.

- As we have seen, shapes can become quite noisy at some iterations of the optimization. Therefore, devising a process for 'smoothing out' shapes could prove to be of interest. As far as this issue is concerned, at first sight, it seems easier to carry out denoising directly on the level set function (i.e. before performing the meshing step for the associated negative subdomain).
- It is very tempting to use the proposed method in combination with a mesh adaptation process. Actually, two different types of mesh adaptation could be considered. At first, a 'geometrical' mesh adaptation method could be devised so that, during the evolution step of the level set function, an increased resolution of the capture of the new shape is possible (see the work [30, 20] in this direction). A different mesh adaptation method, based e.g. on a posteriori error estimates for finite element methods [1], could be aimed at getting a sharper resolution of the linearized elasticity system involved in the computation of a descent direction for the given objective functional.
- It is also very natural to apply this method to other models (still in the framework of shape optimization) which could take advantage of an explicit discretization of the boundary of shapes (geometrical

constraints - e.g. minimum and maximum thickness constraints - are of this nature). Many other mechanical problems could also benefit from the mesh evolution aspect of our method, e.g. fluid-solid interactions.

- Eventually, it could be worth considering a simpler, less costly, mesh deformation method in the spirit of [15, 35, 51] for the last iterations of the optimization procedure, where shapes can be assumed to undergo very small changes from one iteration to the next (in particular, their topology is fixed).

REFERENCES

[1] M. AINSWORTH AND J. T. ODEN, *A posteriori error estimation in finite element analysis*, Comput. Meths. Appl. Mech. Engrg., 142, (1997), pp. 1–88.

[2] G. ALLAIRE, *Conception optimale de structures*, Mathématiques et Applications 58, Springer, Heidelberg (2006).

[3] G. ALLAIRE, E. BONNETIER, G. FRANCFORT AND F. JOUVE, *Shape optimization by the homogenization method*, Numerische Mathematik, 76, (1997), pp. 27–68.

[4] G. ALLAIRE AND C. DAPOGNY, *A linearized approach to worst-case design in parametric and geometric shape optimization*, to appear in M3AS (2014).

[5] G. ALLAIRE, C. DAPOGNY, G. DELGADO AND G. MICHAILIDIS, *Multi-phase optimization via a level set method*, to appear in ESAIM: Control, Optimisation and Calculus of Variations (2014).

[6] G. ALLAIRE, C. DAPOGNY AND P. FREY, *Topology and Geometry Optimization of Elastic Structures by Exact Deformation of Simplicial Mesh*, C. R. Acad. Sci. Paris, Ser. I, vol. 349, no. 17, (2011), pp. 999–1003.

[7] G. ALLAIRE, C. DAPOGNY AND P. FREY, *A mesh evolution algorithm based on the level set method for geometry and topology optimization*, Struct. Multidisc. Optim., 48, (2013), pp. 711–715.

[8] G. ALLAIRE, F. DE GOURNAY, F. JOUVE AND A.M. TOADER, *Structural optimization using topological and shape sensitivity analysis via a level-set method*, Control and Cybernetics, 34 (2005) pp. 59–80.

[9] G. ALLAIRE AND F. JOUVE, *A level-set method for vibration and multiple loads structural optimization*, Comput. Meths. Appl. Mech. Engrg., 194, (2005), pp. 3269–3290.

[10] G. ALLAIRE AND F. JOUVE, *Minimum stress optimal design with the level set method*, Engineering Analysis with Boundary Elements, 32 (2008) pp. 909–918.

[11] G. ALLAIRE, F. JOUVE AND A.M. TOADER, *Structural optimization using shape sensitivity analysis and a level-set method*, J. Comput. Phys., 194 (2004) pp. 363–393.

[12] G. ALLAIRE, F. JOUVE AND N. VAN GOETHEM, *Damage evolution in brittle materials by shape and topological sensitivity analysis,* J. Comput. Phys., 230 (2011) pp. 5010–5044.

[13] G. ALLAIRE AND O. PANTZ, *Structural Optimization with FreeFem++*, Struct. Multidiscip. Optim., 32, (2006), pp. 173–181.

[14] J.F. AUJOL AND G. AUBERT, *Signed distance functions and viscosity solutions of discontinuous Hamilton-Jacobi Equations*, INRIA Technical Report, 4507 (2002).

[15] T. J. BAKER, *Mesh Movement and Metamorphosis*, Eng. Comput. 18, 1, (2002), pp. 188–198.

[16] M.P. BENDSØE AND N. KIKUCHI, *Generating Optimal Topologies in Structural Design using a Homogenization method*, Comp. Meth. Appl. Mech. Eng., 71, (1988), pp. 197–224.

[17] M.P. BENDSØE AND O. SIGMUND, *Topology Optimization, Theory, Methods and Applications, 2nd Edition* Springer Verlag, Berlin Heidelberg (2003).

[18] B. BOURDIN AND A. CHAMBOLLE, *Design-dependent loads in topology optimization*, ESAIM Contr. Optim. Calc. Var., 9, (2003), pp. 19–48.

[19] V. BRAIBANT AND C. FLEURY, *Shape optimal design using B-splines*, Comput. Meths. Appl. Mech. Engrg., 44, (3), (1984), pp. 247–267.

[20] C. BUI, C. DAPOGNY AND P. FREY, *An accurate anisotropic adaptation method for solving the level set advection equation*, Int. J. Numer. Methods in Fluids, Volume 70, Issue 7, pp. 899–922 (2012).

[21] 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.

[22] M. BURGER, B. HACKL AND W. RING, *Incorporating topological derivatives into level set methods*, J. Comp. Phys., 194, 1 (2004) pp. 344–362.

[23] J. CÉA , *Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût*, Math. Model. Num. 20, 3 (1986), pp. 371–420.

[24] A. CHERKAEV AND E. CHERKAEVA , *Principal Compliance and Robust Optimal Design*, Journal of Elasticity, 72 (2003), pp.71–98.

[25] D. CHOPP, *Computing minimal surfaces via level-set curvature flow*, J. Comput. Phys., 106, pp. 77-91 (1993).

[26] A. N. CHRISTIANSEN, M. NOBEL-JØRGENSEN, N. AAGE, O. SIGMUND AND J. A. BÆRENTZEN , *Topology optimization using an explicit interface representation*, to appear in Struct. Multidisc. Optim. (2013).

[27] P.G. CIARLET, *The Finite Element Method for Elliptic Problems*, North Holland Publishing Company, (1978).

[28] C. DAPOGNY, *Thèse de l'Université Pierre et Marie Curie.*

[29] C. Dapogny, C. Dobrzynski and P. Frey, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, submitted (2013).

[30] C. Dapogny, P. Frey, *Computation of the signed distance function to a discrete contour on adapted triangulation*, Calcolo, Volume 49, Issue 3, pp. 193-219 (2012).

[31] J. D. Deaton and R. V. Grandhi, *A survey of structural and multidisciplinary continuum topology optimization: post 2000*, Struct. Multidisc. Optim., (2013) DOI 10.1007/s00158-013-0956-z.

[32] M.C. Delfour and J.-P. Zolesio, *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, SIAM, Philadelphia 2nd ed. (2011).

[33] N. P. van Dijk, K. Maute, M. Langelaar and F. van Keulen, *Level-set methods for structural topology optimization: a review*, Struct. Multidisc. Optim., (2013) DOI 10.1007/s00158-013-0912-y.

[34] C. Dobrzynski, *Adaptation de maillage anisotrope 3d et application à l'aéro-thermique des bâtiments*, Thèse de l'Université Paris VI, (2005).

[35] C. Dobrzynski and P. Frey, *Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations*, Proc. 17th Int. Meshing Roundtable, Pittsburgh, (2008).

[36] P. Duysinx, L. Van Miegroet, T. Jacobs and C. Fleury, *Generalized shape optimization using X-FEM and level set methods*, IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials Solid Mechanics and Its Applications, 137, (2006), pp. 23–32.

[37] H. Eschenauer, V. Kobelev, A. Schumacher, *Bubble method for topology and shape optimization of structures,* Structural Optimization, 8, (1994), pp. 42–51.

[38] L. Freitag and C. Olliver-Gooch, *Tetrahedral Mesh Improvement Using Face Swapping and Smoothing*, Int. J. Numer. Methods in Engineering, 40 (1997), pp. 3979–4002.

[39] P. Frey and H. Borouchaki, *Texel : triangulation de surfaces implicites. Partie I : aspects théoriques.*, INRIA : Technical Report, 3066 (1996).

[40] P.J. Frey and P.L. George, *Mesh Generation : Application to Finite Elements*, Wiley, 2nd Edition, (2008).

[41] S. Garreau and Ph. Guillaume and M. Masmoudi, *The topological asymptotic for PDE systems : the elasticity case*, SIAM J. Control. Optim., 39 (2001) pp. 1756–1778.

[42] P.-L. George, H. Borouchaki, *Back to edge flips in 3 dimensions*, Proc. 12th Int. Meshing Roundtable, Santa Fe, New Mexico, U.S.A., (2003).

[43] F. de Gournay, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization.* SIAM J. on Control and Optim., 45, no. 1, 343–367 (2006).

[44] F. de Gournay, G. Allaire and F. Jouve , *Shape and topology optimization of the robust compliance via the level set method*, ESAIM: Control, Optimization and Calculus of Variations, 14, (2008), pp. 43–70.

[45] S.-H. Ha, S. Cho, *Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh*, Computers and Structures, 86, (2008), pp. 844–868.

[46] F. Hecht, *New development in FreeFem++*, J. Numer. Math. 20 (2012), no. 3-4, pp. 251–265.

[47] A. Henrot and M. Pierre, *Variation et Optimisation de Formes : une Etude Géométrique*, collection Mathématiques et Applications, vol. 48, Springer (2005).

[48] L. Li, M. Y. Wang and P. Wei *XFEM schemes for level set based structural optimization*, Front. Mech. Eng. 7, 4, (2012), pp. 335–356.

[49] W. E. Lorensen and H. E. Cline, *Marching cubes: a high resolution 3d surface construction algorithm*, COMPUTER GRAPHICS, 21, 4 (1987), pp. 163–169.

[50] G. Michailidis, *Thèse de l'École Polytechnique* (in preparation)

[51] M. K. Misztal and J. A. Baerentzen, *Topology Adaptive Interface Tracking Using the Deformable Simplicial Complex*, ACM Trans. Graph. 31, 3, (2012), pp. 1–24.

[52] F. Murat and J. Simon, *Sur le contrôle par un domaine géométrique*, Technical Report RR-76015, Laboratoire d'Analyse Numérique (1976).

[53] S. J. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, (2002).

[54] S. J. Osher and F. Santosa, *Level Set Methods for Optimization Problems Involving Geometry and Constraints I. Frequencies of a Two-Density Inhomogeneous Drum*, J. Comput. Phys., 171 (2001) pp. 272–288.

[55] S.J. Osher and J.A. Sethian, *Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.

[56] P.-O. Persson, *Mesh Generation for Implicit Geometries*, Ph.D. thesis, Department of Mathematics, MIT, (2004).

[57] O. Pironneau, *On optimum profiles in Stokes flow*, J. Fluid Mech. 59 (1973), pp. 117–128.

[58] O. Pironneau, F. Hecht, A. Le Hyaric, *FreeFem++ version 2.15-1*, http://www.freefem.org/ff++/.

[59] O. Pironneau, *The finite element methods for fluids.*, Wiley (1989).

[60] J.A. Sethian, *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry,Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, (1999).

[61] J. A. Sethian and A. Wiegmann, *Structural boundary design via level set and immersed interface methods*, J. Comput. Phys., 163, 2 (2000), pp. 489-528.

[62] J. Sokołowski, A. Żochowski, *Topological derivatives of shape functionals for elasticity systems.* Mech. Structures Mach., 29, no. 3, 331–349 (2001).

[63] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization : Shape Sensitivity Analysis*, Springer Ser. Comput. Math., vol. 10, Springer, Berlin (1992).

[64] J. STRAIN, *Semi-Lagrangian Methods for Level Set Equations*, J. Comput. Phys., 151 (1999) pp. 498–533.

[65] T.T.M. TA, *Thèse de l'Université Pierre et Marie Curie* (in preparation)

[66] L.A. VESE AND T.F. CHAN, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, Int. J. Comput. Vision 50 (3) (2002) 271-293.

[67] A. VLACHOS, J. PETERS, C. BOYD AND J.L. MITCHELL, *Curved PN Triangles*, Symposium on Interactive 3*d* Graphics, (2001), pp 159–166.

[68] M.Y. WANG, X. WANG, D. GUO, *A level set method for structural topology optimization,* Comput. Methods Appl. Mech. Engrg., 192, 227–246 (2003).

[69] Q. XIA, T. SHI, S. LIU, M. Y. WANG, *A level set solution to the stress-based structural shape and topology optimization*, Computers and Structures, 90-91, pp. 55-64 (2012).

[70] S. YAMASAKI, T. NOMURA, A. KAWAMOTO, S. NISHIWAKI, *A level set-based topology optimization method targeting metallic waveguide design problems*, Int. J. Numer. Meth. Engng , 87, pp.844-868 (2011).

[71] O.C. ZIENKIEWICZ AND J.S. CAMPBELL, *Shape optimization and sequential linear programming*, in: R.H. Gallagher and O.C. Zienkiewicz, eds., Optimum Structural Design, Wiley, New York, (1973), pp. 109–126.