10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

A mesh evolution algorithm based on the level set method for geometry and topology optimization

Grégoire Allaire[1], Charles Dapogny[2,3], Pascal Frey[2]

[1] Centre de Mathématiques Appliquées (UMR 7641), École Polytechnique 91128 Palaiseau, France.
[2] UPMC Univ Paris 06, UMR 7598, Laboratoire J.-L. Lions, F-75005 Paris, France.
[3] Renault DREAM-DELT'A Guyancourt, France.

1. Abstract
We discuss an approach for structural optimization which features an exact description of the shapes - i.e. by means of a computational mesh - at each stage of the iterative process, while allowing for large deformations (including topological changes) when it comes to describing surface evolution, by taking advantage of the level set method. The cornerstone of the proposed method is a set of algorithms for switching from one of these descriptions to the other. This notably brings into play a method for generating the signed distance function to a discrete contour on an unstructured computational mesh, and a robust algorithm for meshing the negative subdomain of a scalar 'level set' function, supplied in a discrete fashion.

2. Keywords: Geometry and topology optimization · Level set method · Local mesh modifications.

3. Introduction
Since the seminal works [5] [18], the level set method has proved a very valuable tool in the field of structural optimization. The main idea of the method is to represent the admissible shapes Ω - which are embedded in a fixed 'large' computational domain D - by means of an associated *level set function* $\phi : D \to \mathbb{R}$, such that:

$$\begin{cases} \phi(x) < 0 & \text{if} \quad x \in \Omega, \\ \phi(x) = 0 & \text{if} \quad x \in \partial\Omega, \\ \phi(x) > 0 & \text{if} \quad x \in D \setminus \overline{\Omega}. \end{cases} \tag{1}$$

The evolution of the shape Ω through the optimization process is then deduced from the solution of a Hamilton-Jacobi equation for ϕ [13]. In [5], as well as in almost all other works on the level set method in the context of structural optimization, a fixed mesh of D is used and the exact mechanical analysis, which allows to assess the performance of Ω, is approximated by an analysis held on the whole domain D instead of the sole Ω, using the so-called *Ersatz material approach*. In this context, an explicit meshing of the shape is avoided.

As announced in the previous works [3] [2], we propose to add an extra ingredient to this process, asking that, at each iteration, the shape is exactly meshed, which enables precise mechanical computations. Doing so requires that the mesh of D be *unstructured* and change from one iteration to the next since the shape is *explicitly* discretized as part of the mesh of D. Nevertheless, we still retain the versatility of the level set method when it comes to topological changes.

Such a change in perspectives raises several difficulties: let alone the fact that we can no longer rely on finite difference schemes for, among other things, solving Hamilton-Jacobi equations on D (because the mesh of D is no longer Cartesian and we don't want to use two different meshes, thus relying on projections from one to the other), we need efficient tools to switch from the level set representation of shapes to a meshed representation, and conversely.

The hereafter described method differs from previous works on shape optimization using exact meshes of the shapes, which were - at least to our knowledge - always carried out in two space dimensions. Among other things, in [20], the authors applied a level set method on a fixed background mesh, then moved points of this mesh onto the 0 isoline of the level set function so to obtain a mesh of the associated shape. In [19] and [12], the authors inferred from the knowledge of a level set function a sample set of points associated to the shape, then resorted to the Delaunay algorithm to construct a computational mesh of it. Unfortunately, such methods are difficult to extend to three space dimensions. On the contrary, as we shall see below, our method extends to this context without additional theoretical difficulties (even

though it is considerably more tedious to carry out, especially as regards the part related to local remeshing). Moreover, the use of unstructured meshes enables the use of mesh adaptation techniques [10].

4. Setting of the problem

We consider *shapes*, that is, (smooth enough) bounded domains $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$), filled with a linear elastic isotropic material with Hooke's law A. Each shape Ω is clamped on a given part $\Gamma_D \subset \partial\Omega$, submitted to external loads g on another part $\Gamma_N \subset \partial\Omega$, with $\Gamma_D \cap \Gamma_N = \emptyset$ (without loss of generality, we omit body forces); the complementary part $\Gamma := \partial\Omega \setminus (\Gamma_D \cup \Gamma_N)$ is free. In this context, the displacement field u of the structure is the unique solution to the system the linear elasticity system:

$$\left\{ \begin{array}{cc} -\mathrm{div}\,(Ae(u)) = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ Ae(u).n = g & \text{on } \Gamma_N, \\ Ae(u).n = 0 & \text{on } \Gamma, \end{array} \right. \tag{2}$$

where $e(u) = (\nabla u + \nabla u^T)/2$ is the *linearized strain tensor*, and n is the outer unit normal to $\partial\Omega$.

We aim at minimizing a functional $J(\Omega)$ of the domain, among a set $\mathcal{U}_{ad}$ of *admissible shapes* Ω, such that, among other things $(\Gamma_D \cup \Gamma_N) \subset \partial\Omega$, which leaves only Γ subject to optimization. For the sake of simplicity, we limit ourselves to considering the *compliance* as an objective function. So that the problem is not trivial, a volume constraint is incorporated in the form of a fixed penalization and the considered functional arises as:

$$J(\Omega) = \int_{\Gamma_N} g.u \, ds + \ell \int_\Omega dx, \tag{3}$$

where ℓ is a (positive) Lagrange multiplier associated to the volume constraint.

When it comes to defining a notion of *shape derivative* for such a functional of the domain $J(\Omega)$, we rely on Hadamard's boundary variation method (see [1][16]). In a nutschell, only variations of a given shape Ω of the form $(I + \theta)(\Omega)$ are considered, where $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ is a *small* displacement field. The shape derivative of J at Ω is then defined as the Fréchet differential at 0 of the underlying mapping $\theta \mapsto J((I+\theta)(\Omega))$. The *structure theorem* [9] states that, for a wide class of functionals J, this derivative is of the form:

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \;\; J'(\Omega)(\theta) = \int_\Gamma v \, \theta.n \, ds, \tag{4}$$

for a certain scalar function v on Γ. A *descent direction* for J is then revealed as $-v\,n$. For instance, the shape derivative of (3) is (see [5]):

$$J'(\Omega)(\theta) = \int_\Gamma (\ell - Ae(u) : e(u)) \; \theta.n \, ds. \tag{5}$$

5. A brief description of the proposed method

Let $D \subset \mathbb{R}^d$ be a fixed computational domain which encloses all admissible shapes: as described in [3], we rely on two alternative descriptions of shapes $\Omega \subset D$:

- *The level-set description*: Ω is known as the negative subdomain of a scalar function ϕ in the sense that (1) holds. In the numerical context, ϕ is discretized at the vertices of a simplicial mesh of D.
- *The meshed description*: the whole domain D is equipped with a (conformal) simplicial mesh $\mathcal{T}_\Omega$, a part of which is a mesh of Ω, i.e. the entities (edges, faces, etc...) of a mesh of Ω also belong to $\mathcal{T}_\Omega$.

We first describe how to pass from a meshed domain to a level set representation of a domain.

Let $\mathcal{T}_\Omega$ be a simplicial mesh of D, in which $\Omega \subset D$ is explicitly discretized. In order to generate a level set function ϕ associated to Ω on $\mathcal{T}_\Omega$, we compute an approximation of the *signed distance function* d_Ω to Ω on the *unstructured* mesh $\mathcal{T}_\Omega$, which enjoys crucial properties as regards numerical stability [6], that are in close connection with the unit gradient property of d_Ω:

$$|\nabla d_\Omega| = 1, \text{ a.e. in } \mathbb{R}^d. \tag{6}$$

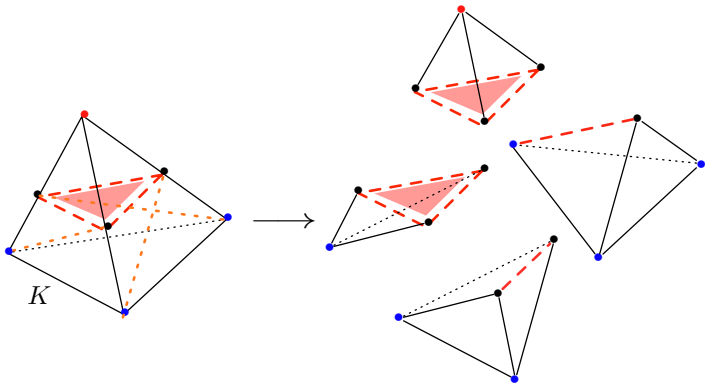

Figure 1: One of the possible configurations when splitting a simplex K along the 0 level set of a function ϕ, which is positive at the red node, negative at the blue ones.

To achieve this, we follow our previous work [8]: starting from *any* level set function $\phi_0 : D \to \mathbb{R}$ associated to Ω (which is easily obtained numerically), equation (6) is turned into the so-called *time-dependent Eikonal equation*:

$$\text{Find } d(t,x) \text{ s.t. } \begin{cases} \dfrac{\partial d}{\partial t} + \text{sgn}(d_0)(||\nabla d|| - 1) = 0 & \forall t > 0, x \in D \\ d(t=0,x) = \phi_0(x) & \forall x \in D \end{cases} \tag{7}$$

which is known to enjoy a unique solution in some sense, whose steady state is precisely d_Ω. An iterative numerical scheme for computing d - thus d_Ω - is then derived from the continuous formula for $d(t,x)$ [6].

We now turn to the description of the meshing process of the negative subdomain of a given level set function.

Let $\mathcal{T}$ be a simplicial mesh of D and ϕ a (piecewise linear) level set function on D, defined at the vertices of $\mathcal{T}$, which accounts for a (polyhedral) domain $\Omega := \{x \in D \,|\, \phi(x) < 0\}$. Note that Ω is not explicitly discretized in $\mathcal{T}$. The proposed method for modifying $\mathcal{T}$ into a new mesh $\widetilde{\mathcal{T}}$ of D, in which Ω is explicitly discretized, involves two steps (see [14] for another interesting approach).

Step 1: Rough discretization of Ω into $\mathcal{T}$. Each simplex $K \in \mathcal{T}$ which is crossed by the 0 level set $\partial\Omega = \{x \in D \,|\, \phi(x) = 0\}$ of ϕ is split into several simplices, in such a way that $K \cap \partial\Omega$ explicitly appears in the resulting mesh. This is a rather easy, purely logical step, which relies on patterns, depending on the relative signs and values of ϕ at the vertices of each such simplex [10] (see figure 1). This step produces an intermediate mesh $\mathcal{T}_{temp}$ of D in which Ω appears as a submesh. Unfortunately, $\mathcal{T}_{temp}$ is bound to be severely *ill-shaped* - i.e. to contain very thin, or flat elements. This is a real problem, since it is well-known that the performances (in terms of accuracy, convergence rate, etc.) of finite element methods when held on a particular mesh strongly depend on its *quality*, that is, grossly speaking, on how close its elements are from the equilateral simplex (triangle in $2d$, tetrahedron in $3d$).

This leaves us with the need for a dramatic improvement in the quality of mesh $\mathcal{T}_{temp}$.

Step 2: Quality-oriented local mesh modifications. This step is the most tedious of the whole process, and is the only one that is fundamentally different from its two-dimensional equivalent. We simply sketch the salient features, referring to [7] for details. From the ill-shaped mesh $\mathcal{T}_{temp}$, a well-shaped mesh $\widetilde{\mathcal{T}}$ is obtained by repeatedly applying four local operations. Each one of them exists under two rather different forms, depending on whether it is applied to a completely internal configuration, or to a situation which lies on the boundary $\partial\Omega$ (resp. ∂D) of the considered implicit domain (resp. computational box).

Here is a short description of these four local remeshing operations [10] (see figure 3 for illustrations in surface configurations).

- *Edge split:* split an edge pq of $\mathcal{T}_{temp}$ which is 'too long', introducing a new vertex m. An edge is said 'too long' when its length is larger than a prescribed size taking into account a user-specified

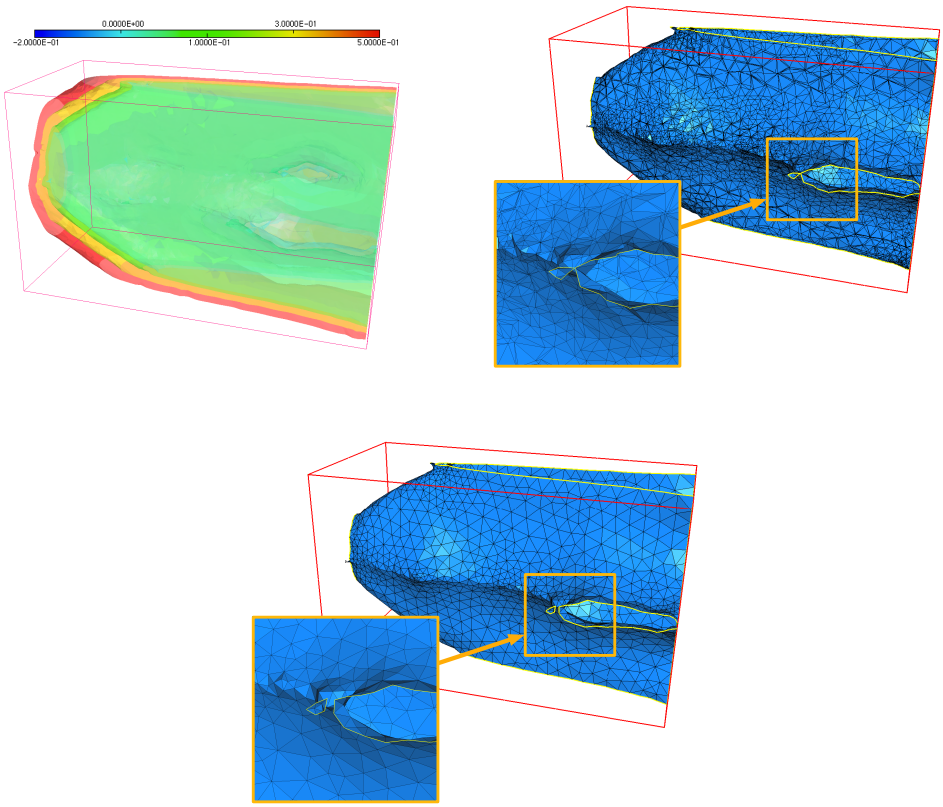

Figure 2: *Top-left:* Isovalues of a level set function ϕ defined over the whole box D; *top-right:* resulting ill-shaped mesh $\mathcal{T}_{temp}$ after the rough discretization of the 0 level set of ϕ; *bottom:* final, well-shaped mesh $\widetilde{\mathcal{T}}$, enclosing a mesh of Ω as a submesh (only $\partial\Omega$ is represented in the last two cases).

size feature and the local curvature of Ω. When pq is an internal edge, m is simply inserted as the midpoint of pq, whereas it should lie on $\partial\Omega$ when pq is a boundary edge. In both cases the connectivities of the mesh are updated accordingly.

- *Edge collapse:* merge the two endpoints of an edge of $\mathcal{T}_{temp}$ whenever it is 'too short'. This operator should be controlled very strictly when applied to an edge $pq \in \partial\Omega$, since it is likely to degrade the geometric approximation of Ω accounted for by $\mathcal{T}_{temp}$, or even to violate the topology of $\partial\Omega$.
- *Edge swap:* change the connectivities of $\mathcal{T}_{temp}$, keeping its vertices' positions unchanged, whenever it helps improving the overall quality of the mesh and does not jeopardize the accuracy of the description of Ω.
- *Vertex relocation:* move a vertex $p \in \mathcal{T}_{temp}$ to a new position $\widetilde{p}$, without altering the connectivities of the mesh, provided it helps improving the overall mesh quality and does not degrade the geometric approximation of Ω.

These operators serve different purposes: whereas the first two are mostly sampling operators, insofar as they are meant to enrich or decimate the mesh so that it ends with a correct density of vertices, the last two are driven so that the quality of the mesh is enhanced. This second step ends with a mesh $\widetilde{\mathcal{T}}$ of D which is amenable for computations (see figure 2 for an example).

Remark 1. It is worth noticing that such a local remeshing algorithm could be used as a post-treatment for topology optimization, when an exact mesh of the optimal shape is sought from the datum of the associated level set function, or even for the hereafter described geometric optimization algorithm, in order to obtain a finer, curvature-dependent mesh of the final shape.

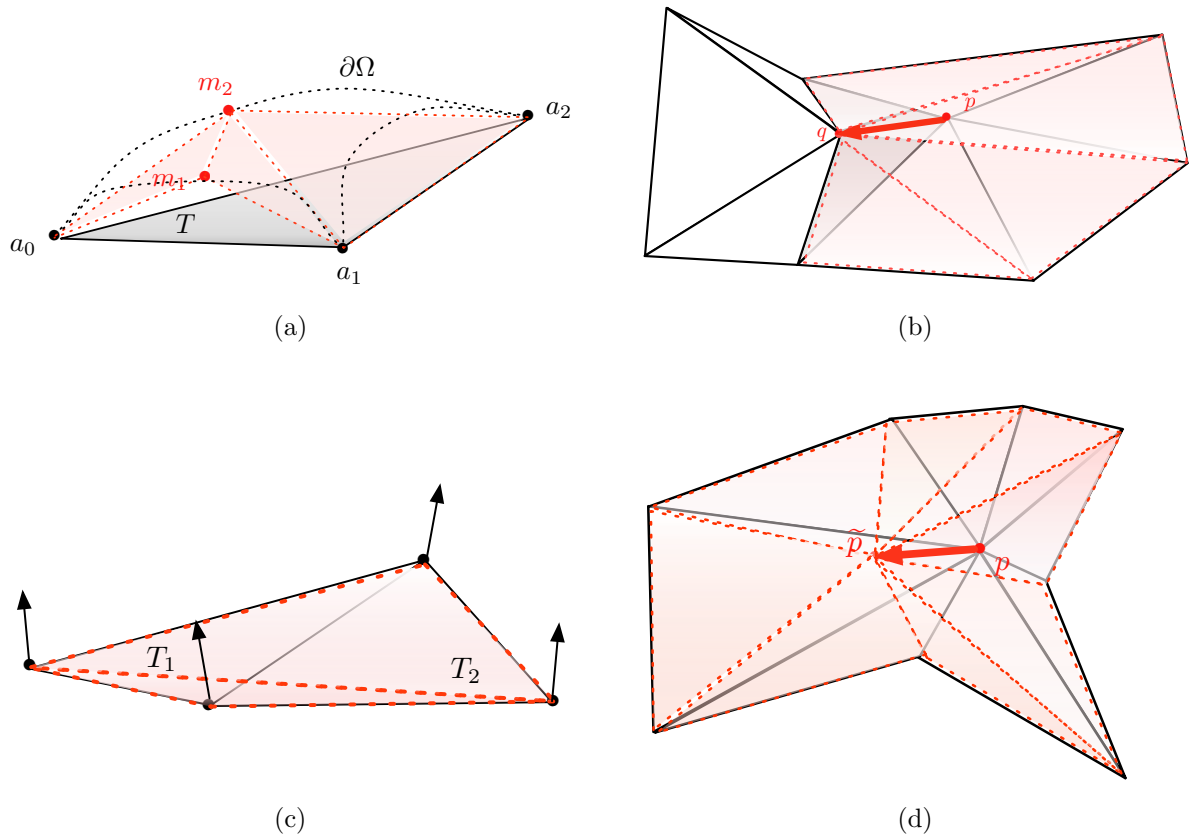

Figure 3: (a): Split of two boundary edges; the new points are inserted on $\partial\Omega$. (b): Collapse of a boundary edge. (c): Swap of a boundary edge. The normal vectors at the four vertices involved are displayed. (d): Relocation of a boundary vertex p, to an 'improving position' $\widetilde{p}$. In all four pictures, the updated boundary triangles during the process are displayed in red transparency.

6. Schematic description of the derived algorithm

Let Ω^0 be an initial shape. Combining the previous tools results in the following algorithm for structural optimization, whose main steps are illustrated on figure 4.

For $n = 0, \ldots$ **till convergence**, start with a shape Ω^n, given by the data of a mesh $\mathcal{T}_{\Omega^n}$ of D, in which Ω^n is explicitly discretized.

1. Retain only the part of $\mathcal{T}_{\Omega^n}$ corresponding to Ω^n, and compute the solution of (2) in Ω^n by a standard finite element method.

2. On the whole unstructured mesh $\mathcal{T}_{\Omega^n}$ of D, generate the signed distance function d_{Ω^n} to Ω^n.

3. Infer from the theoretical formula (5) a descent direction θ^n for (3). This descent direction, which is a priori defined only on the boundary $\partial\Omega^n$ of the current shape is then extended to the whole computational mesh $\mathcal{T}_{\Omega^n}$ owing to a velocity extension procedure such as the one described in [11].

4. Chose a descent step $\tau^n > 0$ and, on mesh $\mathcal{T}_{\Omega^n}$, solve the *level set advection equation*,

$$\begin{cases} \frac{\partial\phi}{\partial t} + \theta^n(x).\nabla\phi = 0 & \text{for } x \in D,\ t \in (0,\tau^n), \\ \phi(0,x) = d_{\Omega^n}(x) & \text{for } x \in D, \end{cases} \tag{8}$$

   as a linear, implicit-in-time approximation of the true nonlinear Hamilton-Jacobi equation for level set evolution, using, for instance, a method of characteristics [17][15]. Remark that, because (8) is solved on the same mesh $\mathcal{T}_{\Omega^n}$ as the one used for the finite element analysis, no projection of the velocity field whatsoever is involved. This yields a level set function $\phi^{n+1} := \phi(\tau^n, .)$ on $\mathcal{T}_{\Omega^n}$ associated to the new shape Ω^{n+1}.

5. Discretize the 0 level set of ϕ^{n+1} in the mesh $\mathcal{T}_{\Omega^n}$ along the lines of section 5 to obtain a new mesh $\mathcal{T}_{\Omega^{n+1}}$ of D, in which Ω^{n+1} is explicitly discretized.

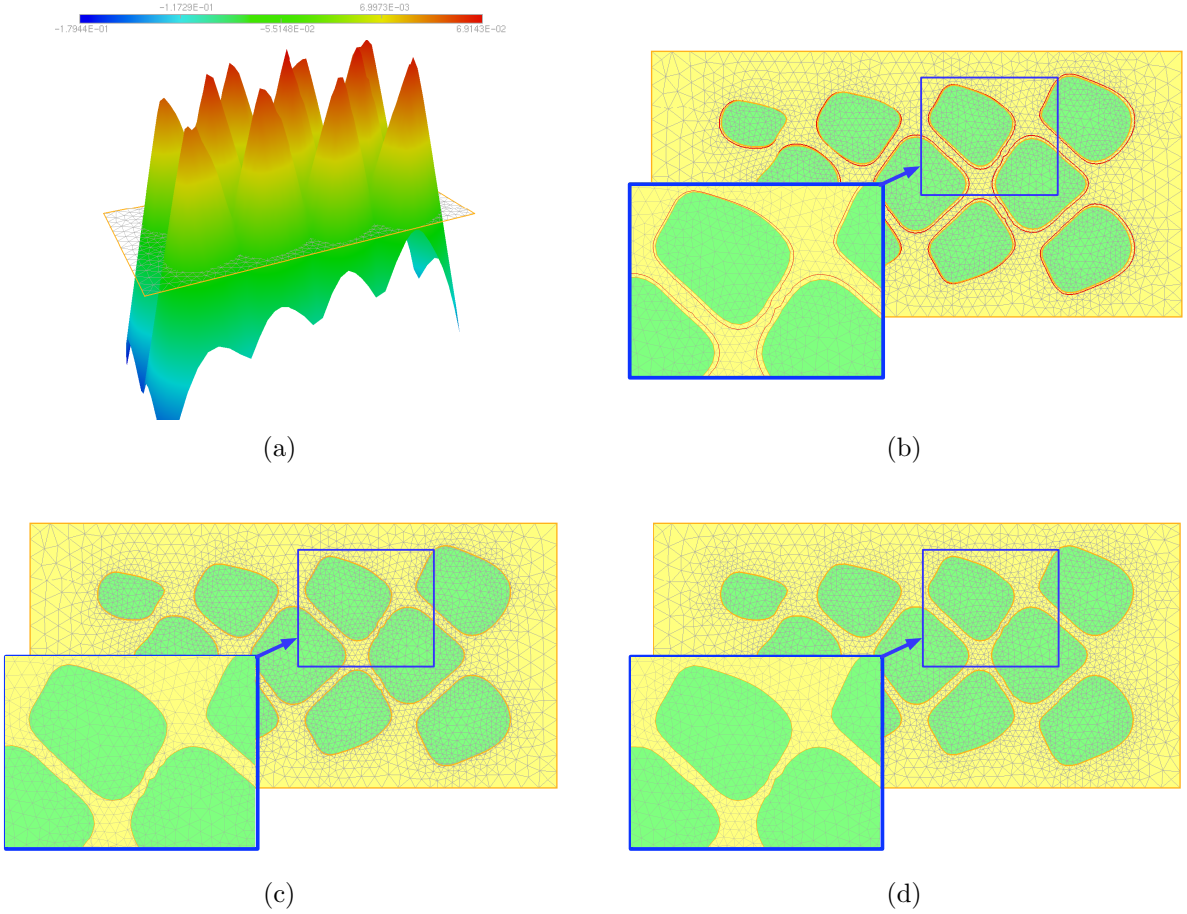

Figure 4: Two-dimensional illustration of the proposed algorithm; (a): at the beginning of the n^{th} iteration, the shape Ω^n is explicitly discretized in the mesh $\mathcal{T}_{\Omega^n}$ of D, and the signed distance function d_{Ω^n} is computed. (b): in red, 0 level set of ϕ^{n+1} on mesh $\mathcal{T}_{\Omega^n}$; in yellow, edges of the shape Ω^n. (c): An intermediate, ill-shaped mesh $\mathcal{T}^n_{temp}$ is obtained, in which Ω^{n+1} is explicitly discretized. (d): $\mathcal{T}^n_{temp}$ is modified into a well-shaped mesh $\mathcal{T}_{\Omega^{n+1}}$ in which Ω^{n+1} is explicitly discretized.

7. Numerical examples

The proposed method is applied to three benchmark test cases in structural optimization. Our first example is the two-dimensional *Optimal Mast* test case, reported on figure 5: a structure, embedded in a T-shaped box of dimensions 80×120 and made of an isotropic elastic material of Young modulus $E = 1$ and Poisson ratio $\nu = 0.3$, is clamped at the two bottom corners of the box, and submitted to two vertical loads $g = -e_z$, applied on the left and right arms of the box. We aim at minimizing the objective function (3) with a fixed Lagrange multiplier $\ell = 1$ as for the volume constraint. We run 100 iterations of the algorithm detailed in section 6; each mesh $\mathcal{T}_{\Omega^n}$ has about 8000 vertices (that is, about 16000 triangles), and the entire computation takes less than 5 minutes on a laptop computer

Our second example is the famous three-dimensional *Cantilever* problem, depicted in figure 6: a structure, embedded in a box of dimensions $2.4 \times 5 \times 3$, , is clamped on the right side, and submitted to a unit vertical load $g = -e_z$ on a small area located at the middle of its left side. The initial shape is the full box, perforated by several holes, and the Lagrange multiplier for the volume constraint is set to $\ell = 0.05$. We run 80 iterations of the above algorithm; each mesh $\mathcal{T}_{\Omega^n}$ enjoys about $16,000$ vertices (say 90000 tetrahedra), and the entire computation takes about an hour on a laptop computer. Note that our algorithm has been able to change dramatically the topology and yet shapes are exactly meshed at each

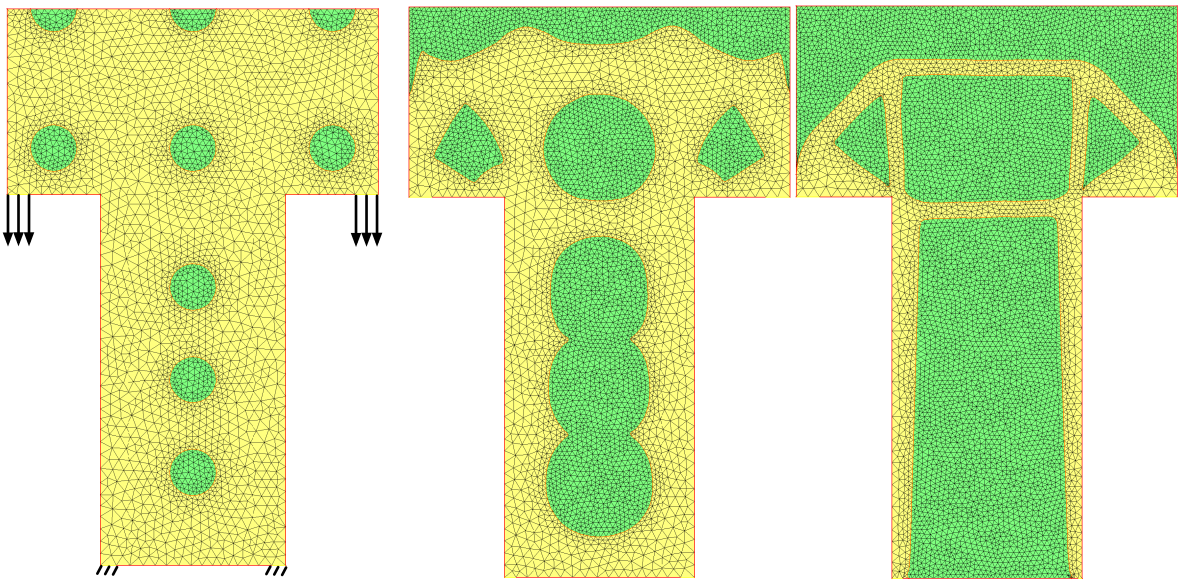

Figure 5: *(From left to right)* Initial (together with boundary conditions), 30^{th} and final iterations of the Optimal Mast test case. The part of each mesh $\mathcal{T}_{\Omega^n}$ corresponding to Ω (resp. $D \setminus \overline{\Omega}$) is displayed in yellow (resp. green).

iteration.

We then turn to our last example, namely that of the three-dimensional *Optimal Bridge* problem (see figure 7): an elastic structure, embedded in a box of dimensions $40 \times 200 \times 50$, is clamped on two areas in its inferior part, and submitted to vertical loads $g = -e_z$, applied on the whole superior part. The objective functional to be minimized is still (3), where ℓ is set to 100. 70 iterations of the optimization procedure described in section 6 are performed. Each mesh $\mathcal{T}_{\Omega^n}$ enjoys about 9000 vertices, and the whole computation takes roughly 45 min. Once again, the topology of the evolving shape changes altogether in the course of the process. As an illustration to remark 1, the final shape is then post-treated into a fine, curvature-dependent mesh ($\approx$ 70000 vertices), as a first step towards reverse engineering (see figure 8).

8. Acknowledgements

This work has been supported by the RODIN project (FUI AAP 13). G. A. is a member of the DEFI project at INRIA Saclay Ile-de-France.

9. References

[1] G. Allaire, Conception optimale de structures, *Collection: Mathématiques et Applications*, vol. 58, Springer (2007).

[2] G. Allaire, C. Dapogny, P. Frey, A mesh evolution algorithm based on the level set method for geometry and topology optimization, accepted in *Struct Multidiscipl Opti*, (2013).

[3] G. Allaire, C. Dapogny, 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, pp. 999-1003 (2011).

[4] G. Allaire, F. Jouve, A.M. Toader, A level-set method for shape optimization, *C. R. Acad. Sci.*, Paris, Série I, 334, pp. 1125-1130 (2002).

[5] G. Allaire, F. Jouve, A.M. Toader, Structural optimization using shape sensitivity analysis and a level-set method, *J. Comput. Phys.*, 194, pp. 363-393 (2004).

[6] J.-F. Aujol, G. Auber, Signed distance functions and viscosity solutions of discontinuous Hamilton-Jacobi Equations, *INRIA Technical Report*, 4507 (2002).

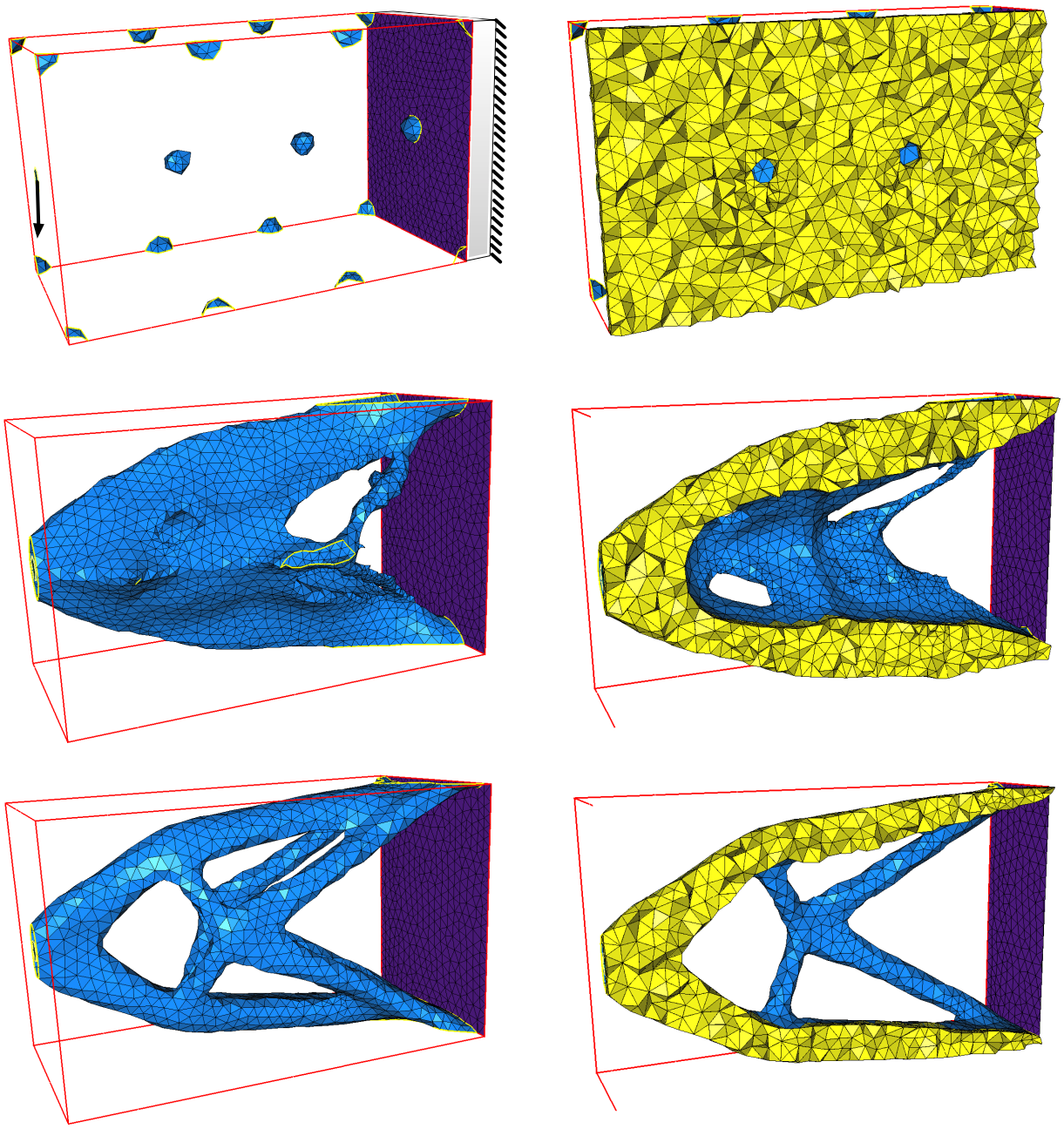

Figure 6: *(From top to bottom)* Initial (together with boundary conditions), 30^{th} and final iterations of the Cantilever test case. Only the boundary $\partial\Omega$ of each shape Ω^n is displayed in the left column, and only the interior part of each mesh $\mathcal{T}_{\Omega^n}$ is displayed in the right column.

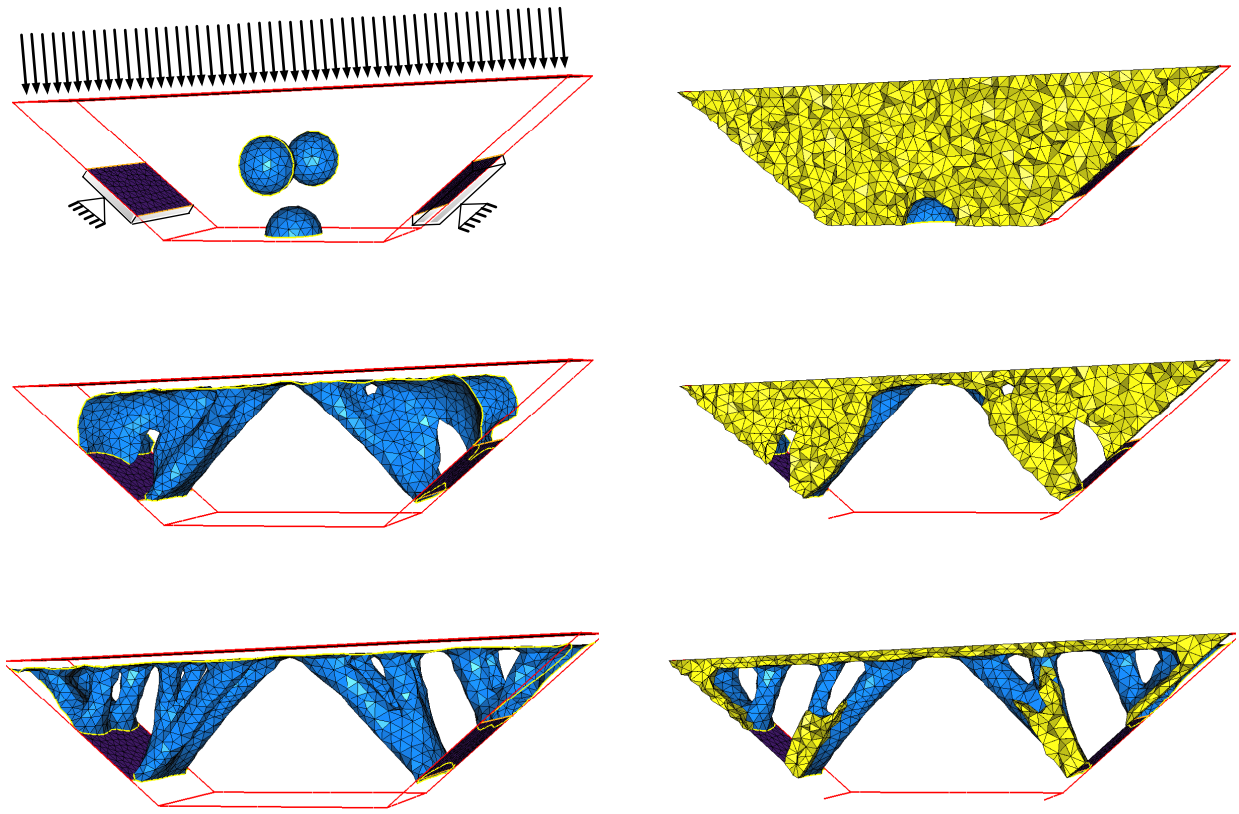

Figure 7: *(From top to bottom)* Initial (together with boundary conditions), 30^{th} and final iterations of the Optimal bridge test case

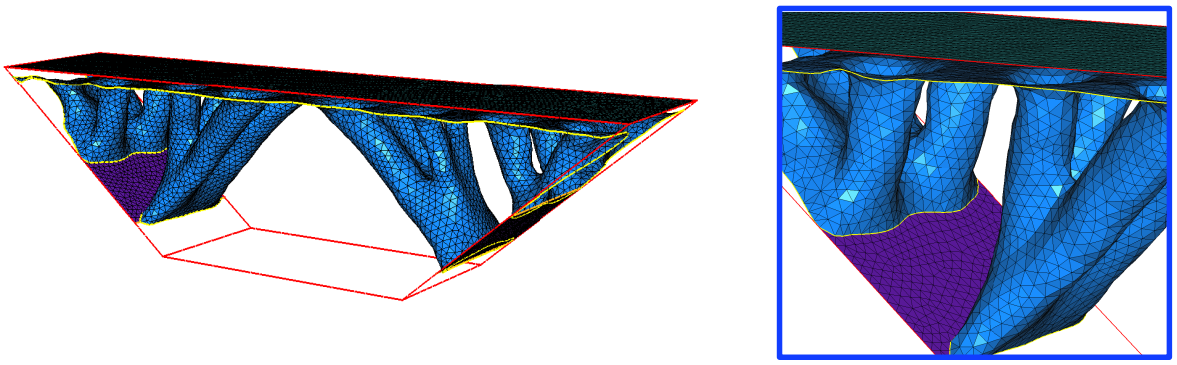

Figure 8: Post-processing of the resulting shape in the Optimal bridge example; only the mesh of the boundary $\partial\Omega$ of the shape is shown.

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

[**7**] C. Dapogny, PhD thesis, *In preparation.*

[**8**] 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).

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

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

[**11**] 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).

[**12**] 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

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

[**14**] P.-O. Persson and G. Strang, A Simple Mesh Generator in MATLAB, *SIAM Review*, 46, no. 2, (2004), pp. 329–345 .

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

[**16**] J. Sokolowski, J.-P. Zolesio, Introduction to shape optimization: shape sensitivity analysis, *Springer Series in Computational Mathematics*, vol 10, Springer, Berlin (1992).

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

[**18**] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Comput. Methods. Appl. Mech. Engrg*, 192, pp. 227-246 (2003).

[**19**] 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).

[**20**] 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).