A shape optimization method, using a level-set based mesh evolution strategy

G. Allaire[1], Ch. Dapogny[1,2,3], and P. Frey[2]

[1] CMAP, UMR 7641 École Polytechnique, Palaiseau, France
[2] Laboratoire J.L. Lions, UPMC, Paris, France
[3] Technocentre Renault, Guyancourt

Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
   - Initializing level set functions
   - Advection of level set functions
3. The proposed algorithm
4. Numerical results

Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
   - Initializing level set functions
   - Advection of level set functions
3. The proposed algorithm
4. Numerical results

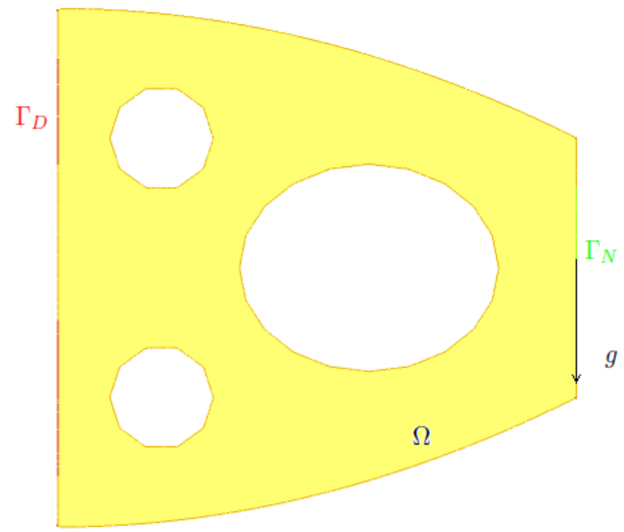
A model problem in linear elasticity

A structure is represented by a bounded open domain $\Omega \subset \mathbb{R}^d$, fixed on a part $\Gamma_D \subset \partial\Omega$ of its boundary, and submitted to a load case g (and no body force), to be applied on $\Gamma_N \subset \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$.

The displacement vector field $u_\Omega : \Omega \to \mathbb{R}^d$ is governed by the linear elasticity system :

$$\begin{cases} -div(Ae(u_\Omega)) = 0 & \text{in } \Omega \\ u_\Omega = 0 & \text{on } \Gamma_D \\ Ae(u_\Omega).n = g & \text{on } \Gamma_N \\ Ae(u_\Omega).n = 0 & \text{on } \Gamma := \partial\Omega \setminus (\Gamma_D \cup \Gamma_N) \end{cases},$$

where $e(u) = \frac{1}{2}({}^t\nabla u + \nabla u)$ is the strain tensor field, $Ae(u) = 2\mu e(u) + \lambda tr(e(u))I$ is the stress tensor, and λ, μ are the Lamé coefficients of the material.



A 'Cantilever'

The deformed cantilever

A model problem in linear elasticity

goal : Given an initial structure Ω_0, find a new domain Ω that minimizes a certain functional of the domain $J(\Omega)$, under a volume constraint.

Example : The work of the external loads g or compliance $c(\Omega)$ of domain Ω :

$$c(\Omega) = \int_\Omega Ae(u_\Omega) : e(u_\Omega)dx = \int_{\Gamma_N} g.u_\Omega \, ds$$

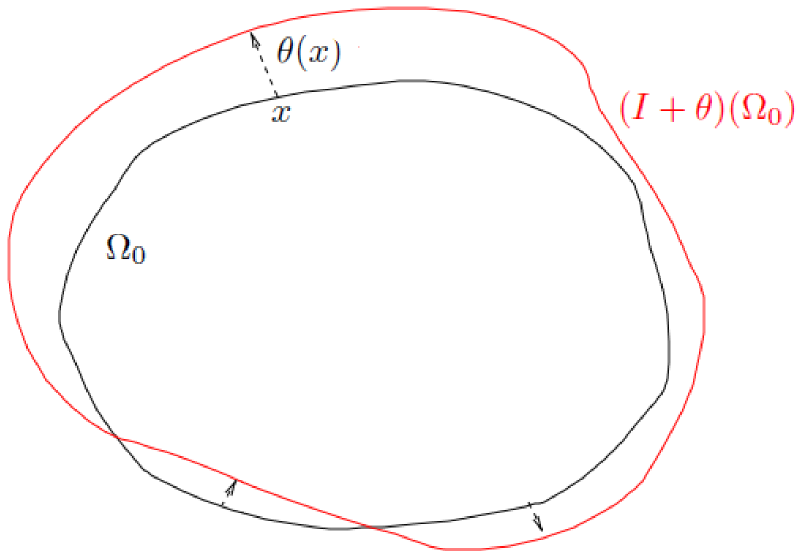
The volume constraint is enforced with a fixed penalty parameter l :

$$\Rightarrow \text{minimize } J(\Omega) := c(\Omega) + l \, Vol(\Omega).$$

Differentiation with respect to the domain : Hadamard's method

Given a reference (initial), smooth domain Ω_0, we parametrize shapes by variations of the form :

$$\Omega_0 \to (I+\theta)(\Omega_0), \quad \theta \in W^{1,\infty}\left(\mathbb{R}^d, \mathbb{R}^d\right).$$



Definition 1 *The shape differential of function $\Omega \mapsto F(\Omega)$ at Ω_0 is the Fréchet-differential of F at 0 of*

$$W^{1,\infty}\left(\mathbb{R}^d, \mathbb{R}^d\right) \ni \theta \mapsto F((I+\theta)(\Omega_0)),$$

Theorem 1 *Ω being a smooth domain, if $g \in H^2(\mathbb{R}^d)$, the above functional J is shape differentiable at Ω and its shape gradient reads :*

$$dJ(\Omega)(\theta) = \int_\Gamma \left(-Ae(u_\Omega) : e(u_\Omega)\right) \theta.n \, ds$$

Differentiation with respect to the domain : Hadamard's method

- This shape gradient provides plenty many natural descent directions for functional J : *for instance*, defining θ as

$$\theta = (Ae(u_\Omega) : e(u_\Omega))\, n$$

yields, for $t > 0$ sufficiently small (*to be found numerically*) :

$$J((I + t\theta)(\Omega)) = J(\Omega) - t \int_\Gamma (\theta.n)^2\, ds + o(t) < J(\Omega)$$

- Note that all the shapes obtained during the process are (at least theoretically speaking) diffeomorphic to the initial one Ω ; hence, no hole can appear, whereas it could be highly beneficial ; a notion of topological gradient has been devised to study the behaviour of a shape with respect with the nucleation of a small hole near each of its points.

The generic numerical algorithm

Gradient algorithm : For $n = 0, \ldots$ until convergence,

1. Compute the solution u_{Ω^n} of the above elasticity system of Ω^n.
2. Compute the shape gradient $dJ(\Omega^n)$ thanks to the above formula, and infer a descent direction θ^n for the cost functional.
3. Advect the shape Ω^n according to this displacement field, so as to get Ω^{n+1}.

Problem : We need to

- efficiently advect the shape Ω^n at each step
- be able to perform finite element computations on Ω^n at each step, to get u_{Ω^n}, which at first glance requires a mesh of this shape.

The level set method of Allaire-Jouve-Toader

- All the shapes Ω^n are embedded in a fixed computational box $\mathcal{D}$ which is meshed once and for all.

- The successive shapes Ω^n are accounted for in the level set framework, i.e. by the knowledge of a function ψ^n defined on the whole box $\mathcal{D}$ which implicitly defines them.

- At each step n, the exact linear elasticity system on Ω^n is approximated by the Ersatz material approach : the void $\mathcal{D} \setminus \Omega^n$ is filled with a very 'soft' material, which leads to an approximate linear elasticity system, defined on $\mathcal{D}$.

- This approach is very versatile and does not require an exact mesh of the shapes at each iteration.

The proposed method

We propose a slightly different approach which still benefits from the versatility of level set methods to account for large deformations of shapes (even topological changes), but enjoys at each step the knowledge of a mesh of the shape.

- At each step, the shape Ω^n is equipped with an unstructured mesh $\mathcal{T}^n$ when it comes to finite element computations, and is considered through an associated level set function ϕ^n, defined on a larger unstructured computational mesh when dealing with advection of the shape

$$(\Omega^n, \mathcal{T}^n) \to (\Omega^{n+1}, \mathcal{T}^{n+1}) \quad \Leftrightarrow \quad \phi^n \to \phi^{n+1}$$

- The connection between those two ways of describing shapes is made through an unstructured mesh of the computational box $\mathcal{D}$, which is allowed to evolve so that at each step n, the shape Ω^n is explicitly discretized.
  - Level set methods are performed on this unstructured mesh to account for the advection of the shapes $\phi^n \to \phi^{n+1}$.
  - Finite element computations are performed on the part on this mesh corresponding to the shape.

The proposed method

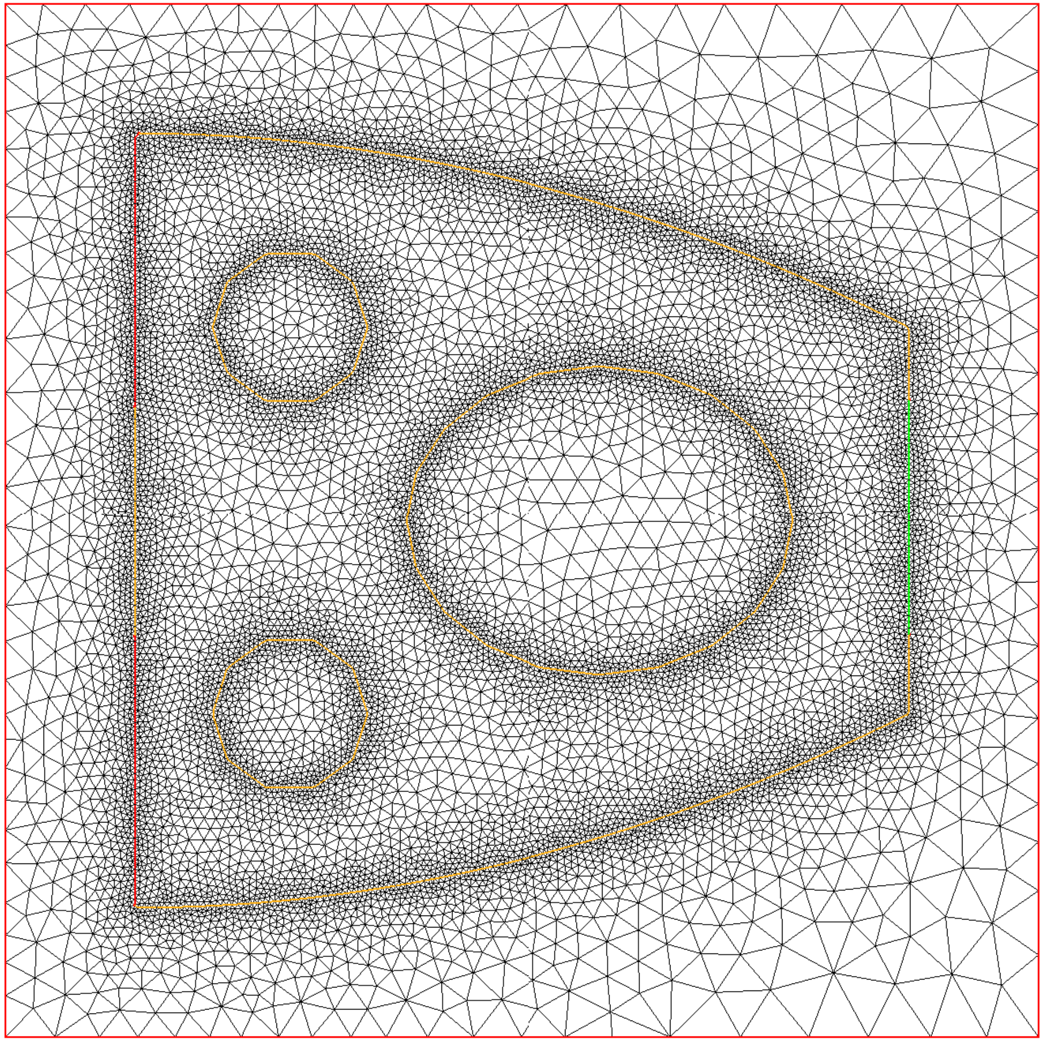


Figure 1: *Shape equipped with a mesh, conformally embedded in a mesh of the computational box.*

Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
    - Initializing level set functions
    - Advection of level set functions
3. The proposed algorithm
4. Numerical results

A few words about the level set Method

A paradigm : *When you want to describe a surface evolution, represent it with an implicit function.*

Given a bounded domain $\Omega \subset \mathbb{R}^d$, define it with a function ϕ on the whole $\mathbb{R}^d$ such that

$$\phi(x) \quad < 0 \quad \text{if } x \in \Omega \quad ; \quad \phi(x) \quad = 0 \quad \text{if } x \in \partial\Omega \quad ; \quad \phi(x) \quad > 0 \quad \text{if } x \in {}^c\Omega$$

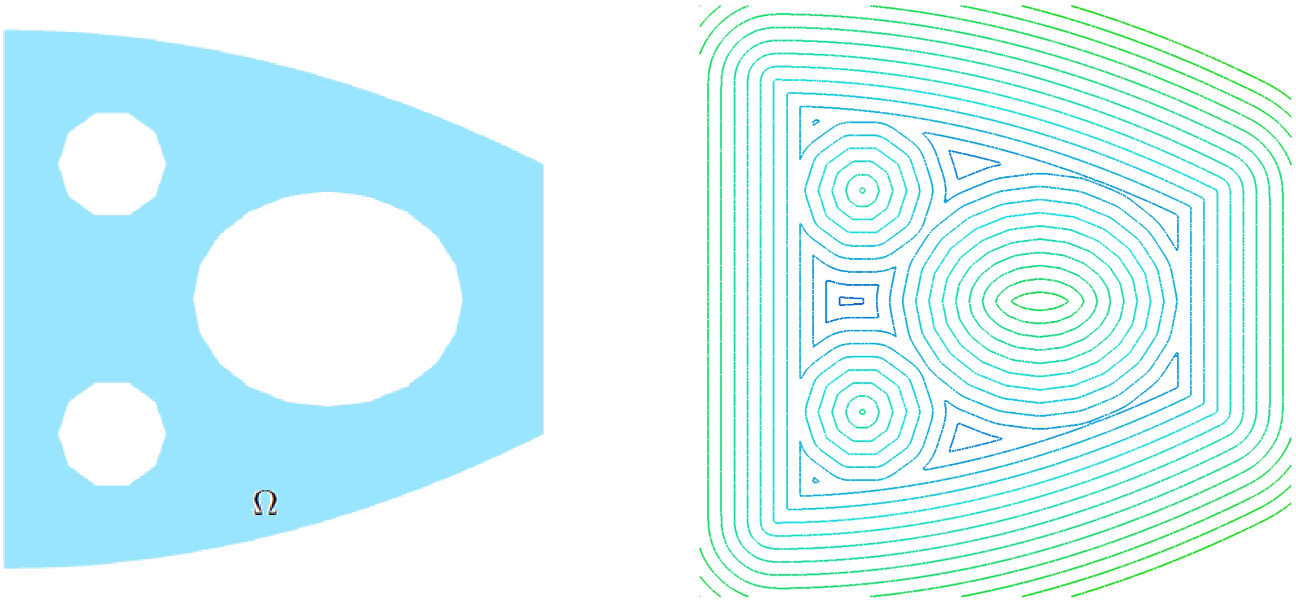


Figure 2: *A bounded domain $\Omega \subset \mathbb{R}^2$ (left), some level sets of an implicit function representing Ω (right).*

Surface evolution equations in the level set framework

Suppose that, for every time t, the domain $\Omega(t) \subset \mathbb{R}^d$ is represented by an implicit function $\phi(t, .)$ on $\mathbb{R}^d$, and is subject to an evolution defined by velocity $v(t,x) \in \mathbb{R}^d$. Then

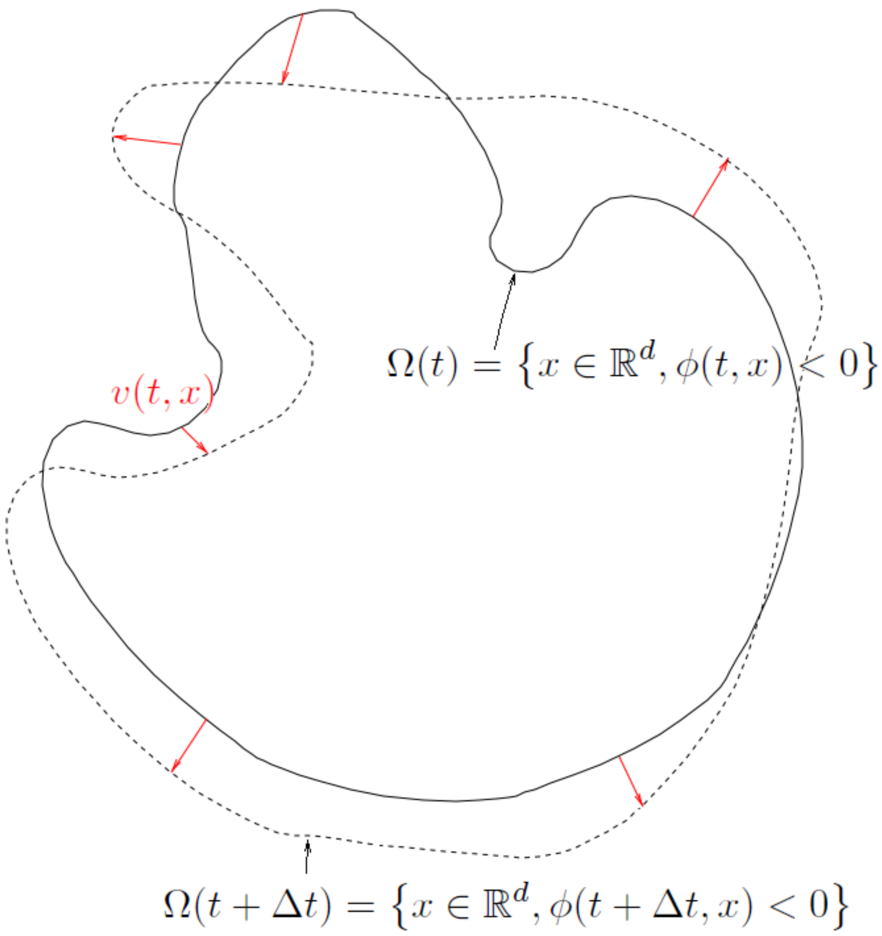
$$\boxed{\forall t,\ \forall x \in \mathbb{R}^d,\ \frac{\partial \phi}{\partial t}(t,x) + v(t,x).\nabla\phi(t,x) = 0}$$

In many applications, the velocity $v(t,x)$ is normal to the boundary $\partial\Omega(t)$:

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

Then the evolution equation rewrites as a Hamilton-Jacobi type equation

$$\boxed{\forall t,\ \forall x \in \mathbb{R}^d,\ \frac{\partial \phi}{\partial t}(t,x) + V(t,x)||\nabla\phi(t,x)|| = 0}$$



Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
    - Initializing level set functions
    - Advection of level set functions
3. The proposed algorithm
4. Numerical results

Initializing level-set functions with the signed distance function

Definition 2 *Let $\Omega \subset \mathbb{R}^d$ a bounded domain. The signed distance function to Ω is the function $\mathbb{R}^d \ni x \mapsto u_\Omega(x)$ defined by :*

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

, where $d(\cdot, \partial\Omega)$ is the usual Euclidean distance

- The signed distance function to a domain $\Omega \subset \mathbb{R}^d$ is the 'canonical' way to initialize an associated level set function : it enables good approximations of $n(x)$, $\kappa(x)$,... and decreases numerical instabilities related to 'bad localization' of the domain, owing to its property of unitary gradient.

- We present here a PDE-based method, working in any dimension, on any simplicial mesh for computing the signed distance function to Ω that dates back to [Chopp] (see also [Sethian] or [Zhao] for different approaches).

The signed distance function as the steady state of a PDE

Suppose $\Omega \subset \mathbb{R}^d$ is implicitly known as

$$\Omega = \left\{x \in \mathbb{R}^d; u_0(x) < 0\right\} \ \ and \ \ \partial\Omega = \left\{x \in \mathbb{R}^d; u_0(x) = 0\right\},$$

where u_0 is a function we only suppose continuous. Then the function u_Ω can be considered as the steady state of the so-called unsteady Eikonal equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + sgn(u_0)(||\nabla u|| - 1) = 0 & \forall t > 0, x \in \mathbb{R}^d \\ u(t = 0, x) = u_0(x) & \forall x \in \mathbb{R}^d \end{cases} \tag{1}$$

The proposed algorithm

THEOREM 2 *Define function* u*,* $\forall x \in \mathbb{R}^d$, $\forall t \in \mathbb{R}_+$,

$$u(t,x) = \begin{cases} sgn(u_0(x)) \inf\limits_{||y|| \leq t} (sgn(u_0(x))u_0(x+y)+t) & if\, t \leq d(x,\partial\Omega) \\ sgn(u_0(x))d(x,\partial\Omega) & if\, t > d(x,\partial\Omega) \end{cases} \tag{2}$$

Let $T \in \mathbb{R}_+$. *Then* u *is the unique uniformly continuous viscosity solution of (1) such that, for all* $0 \leq t \leq T$, $u(t,x) = 0$ *on* $\partial\Omega$.

Idea : *Compute iteratively the solution* $u(t,x)$*, using the exact formula.*

Let dt a small time step, and denote $t^n = ndt$. This formula can be made iterative, denoting $u^n(x) = u(t^n, x)$, we have, for $n = 0, \ldots$

$$\forall x \in {}^c\Omega,\; u^{n+1}(x) = \inf_{||y|| \leq dt} u^n(x+y) + dt$$

$$\forall x \in \Omega,\; u^{n+1}(x) = \sup_{||y|| \leq dt} u^n(x+y) - dt$$

A geometric intuition of the proposed algorithm

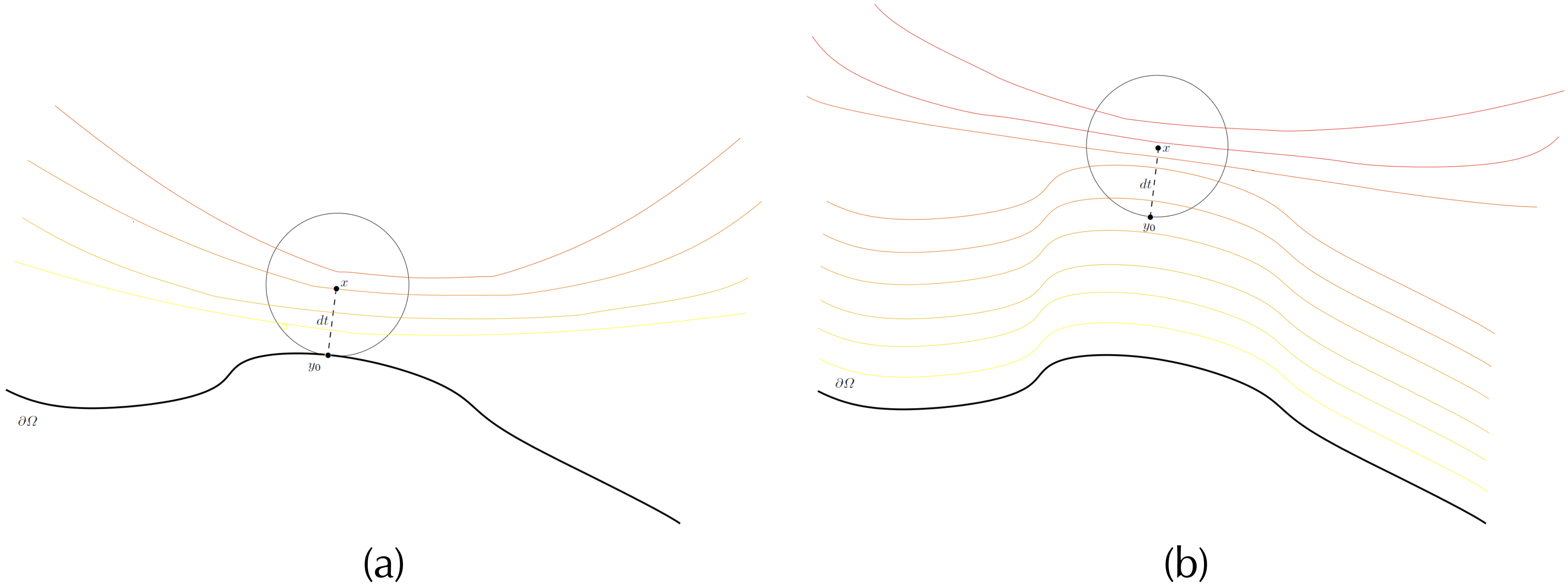


Figure 3: At a given iteration n, the proposed numerical scheme amounts to 'regularize' the value of u^n at point x from its value at point y_0 such that $u^n(y_0) = \inf_{y \in B(x,dt)} u^n(y)$ with the property of unitary gradient, (a) e.g. for a point x at distance dt from $\partial\Omega$, $u^1(x) = u_0(y_0) + dt = dt = d(x, \partial\Omega)$. (b) The property of unit gradient 'propagates' from the boundary $\partial\Omega$, near which values of u^n are 'regularized' at an early stage.

A 2d computational example

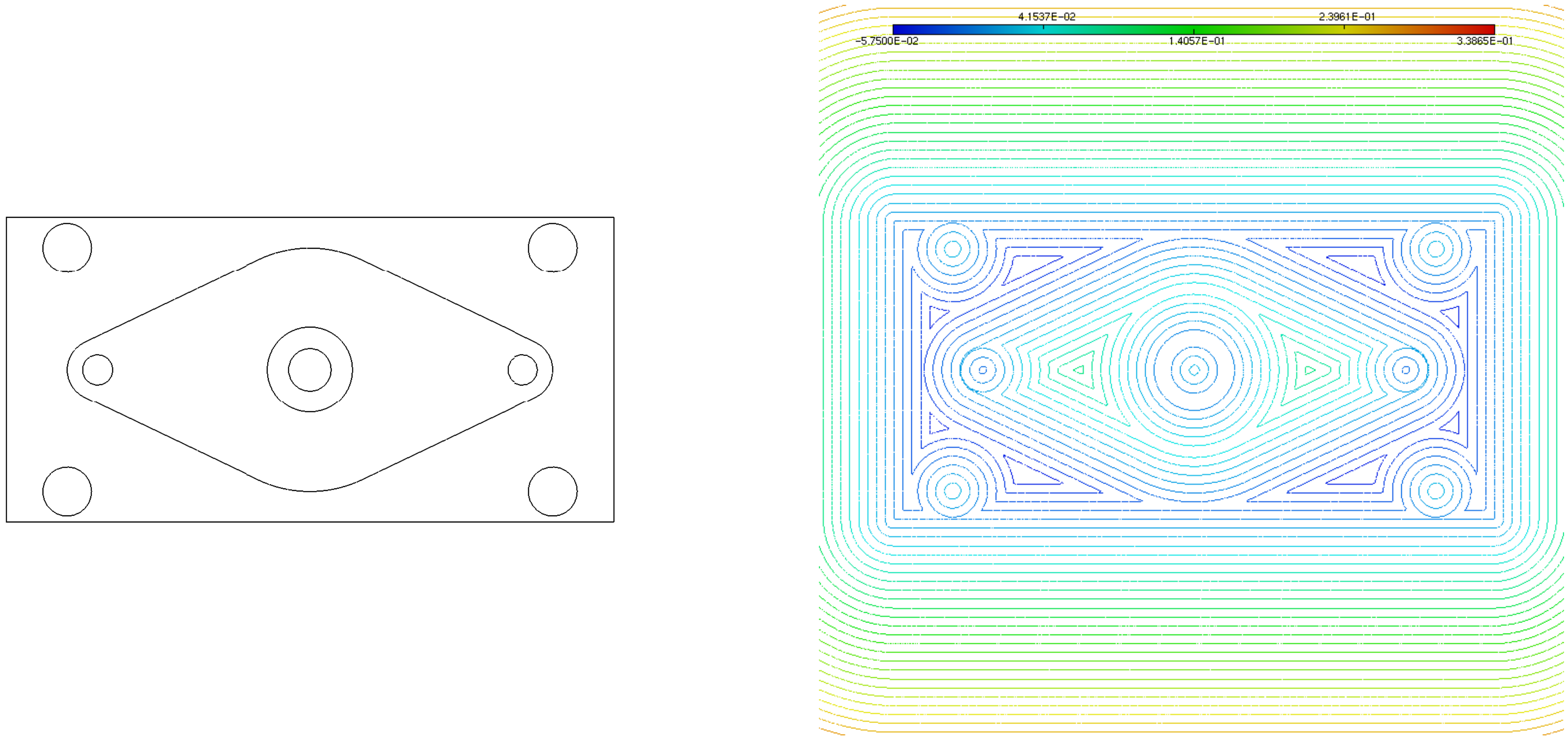


Figure 4: *Computation of the signed distance function to a discrete contour (left), on a fine background mesh (≈ 250000 vertices).*

Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
   - Initializing level set functions
   - Advection of level set functions
3. The proposed algorithm
4. Numerical results

Solving the advection equation with the method of characteristics

We consider the advection equation of a scalar value $\phi(t,.)$ over an time period $\left[t^n, t^{n+1}\right]$ - typically a level set function :

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t,x) + v(t,x).\nabla\phi(t,x) = 0 & \text{for } (t,x) \in (t^n, t^{n+1}) \times \mathbb{R}^d \\ \phi(t^n, x) = \phi^n(x) & \text{if } x \in \mathbb{R}^d \end{cases},$$

where ϕ^n is the (known) scalar value at time t^n.

We are especially interested in the 0-level set of the advected function ϕ, and therefore need to discretize it as a continuous function on the computational domain ($\Rightarrow$ excludes several finite volume methods, or discontinuous Galerkin methods), e.g. a $\mathbb{P}^1$ finite element function.

'Classical' finite element methods are known to behave poorly to solve this equation and following an original idea of [Pironneau], [Strain], we use the method of characteristics.

Solving the advection equation with the method of characteristics

The characteristic curve emerging from point $x \in \mathbb{R}^d$ at time $t \in (t^n, t^{n+1}]$ is the solution $s \mapsto X(s,t,x)$ to the ODE, for $t^n < s < t$:

$$\begin{cases} \dfrac{dX}{dt}(s,t,x) & = & v(s, X(s,t,x)) \\ X(t,t,x) & = & x \end{cases},$$

and the solution to the advection equation is provided by the following formula

THEOREM 3 *Let* $v : \left[t^n, t^{n+1}\right] \times \mathbb{R}^d \to \mathbb{R}^d$ *be of class* $\mathcal{C}^1$*, and assume there exists a constant* $\kappa > 0$ *such that*

$$\forall (t,x) \in \left[t^n, t^{n+1}\right] \times \mathbb{R}^d, \; ||v(t,x)|| \leq \kappa(1 + ||x||)$$

Then if the initial state ϕ^n *is of class* $\mathcal{C}^1$*, the above advection equation admits a unique* $\mathcal{C}^1$ *solution over* $\mathbb{R}^d$*, which is*

$$\boxed{\forall x \in \mathbb{R}^d, \; \phi(t^{n+1}, x) = \phi^n(X(t^n, t^{n+1}, x))}$$

Solving the advection equation with the method of characteristics

- Each function $\phi(t^n, .)$ is approximated by means of a $\mathbb{P}^1$-finite element function.

- Given a computed approximation $\widetilde{\phi^n}$ of $\phi(t^n, .)$, one solves, for each node x of the computational mesh, the ODE for $s \mapsto X(s, t^{n+1}, x)$, thanks to a 4^{th} order Runge-Kutta scheme, and in particular get the foot of the characteristic line $X(t^n, t^{n+1}, x)$.

- The required approximation $\widetilde{\phi^{n+1}}$ for $\phi(t^{n+1}, .)$ is then obtained, from the exact formula, as the $\mathbb{P}^1$-finite element function such that for each node x of the mesh :

$$\widetilde{\phi^{n+1}}(x) = \widetilde{\phi^n}(X(t^n, t^{n+1}, x))$$

- Convergence results can be quite easily obtained for this numerical scheme. It turns out to be quite slow, but can be accelerated with higher-order spatial discretization.

Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
   - Initializing level set functions
   - Advection of level set functions
3. The proposed algorithm
4. Numerical results

Numerical implementation

- At each iteration, the shape Ω^n is endowed with an unstructured mesh $\mathcal{T}^n$ of a larger, fixed, bounding box $\mathcal{D}$, in which a mesh of Ω^n explicitly appears as a submesh.

  $\Rightarrow$ Then, at each iteration, both Ω and ${}^c\overline{\Omega}$ are exactly meshed.

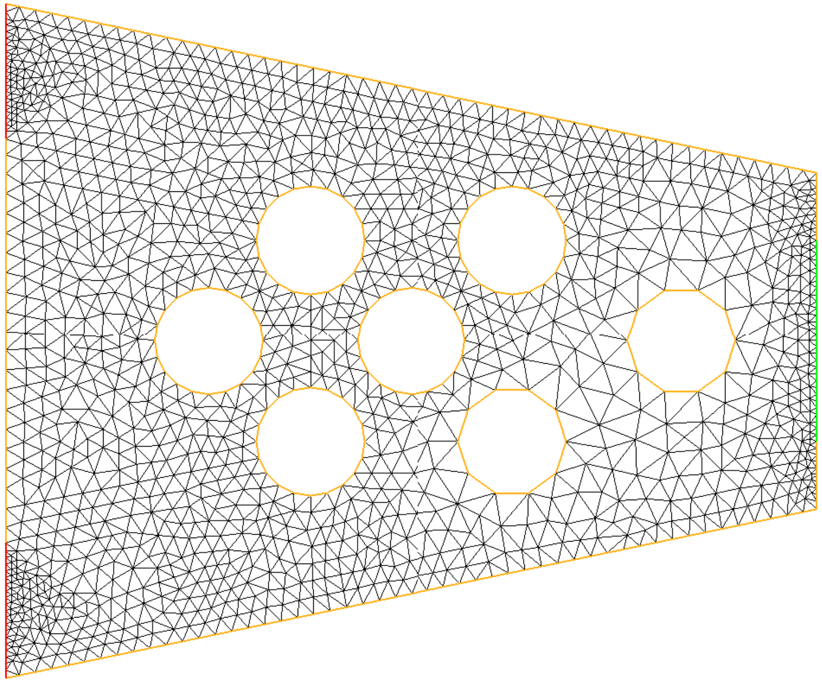
- When dealing with finite element computations on Ω^n, the part of $\mathcal{T}^n$, exterior to Ω^n is simply 'forgotten'.

- When dealing with the advection step, a level set function ϕ^n is generated on the whole mesh $\mathcal{T}^n$, and the level set advection equation is solved on this mesh, to get ϕ^{n+1}.

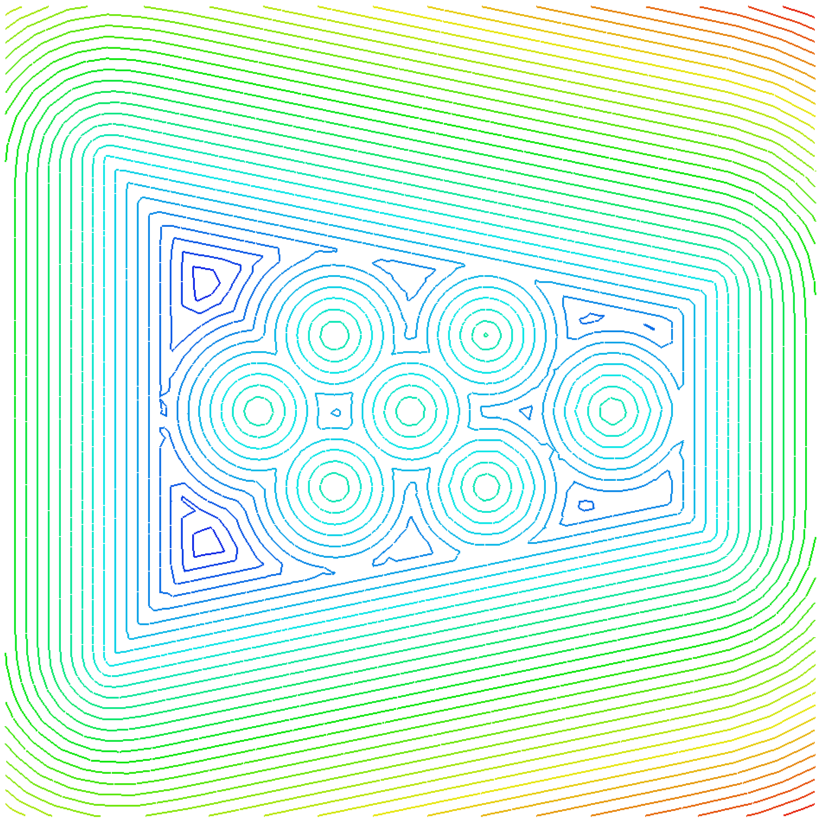
- From the knowledge of ϕ^{n+1}, a new unstructured mesh $\mathcal{T}^{n+1}$, in which the new shape Ω^{n+1} explicitly appears, is recovered, discretizing the new shape in the previous mesh $\mathcal{T}^n$.

The algorithm in motion

Start with an initial shape Ω_0, and generate its signed distance function over a computational domain $\mathcal{D}$, equipped with an unstructured mesh.



(a) The initial shape

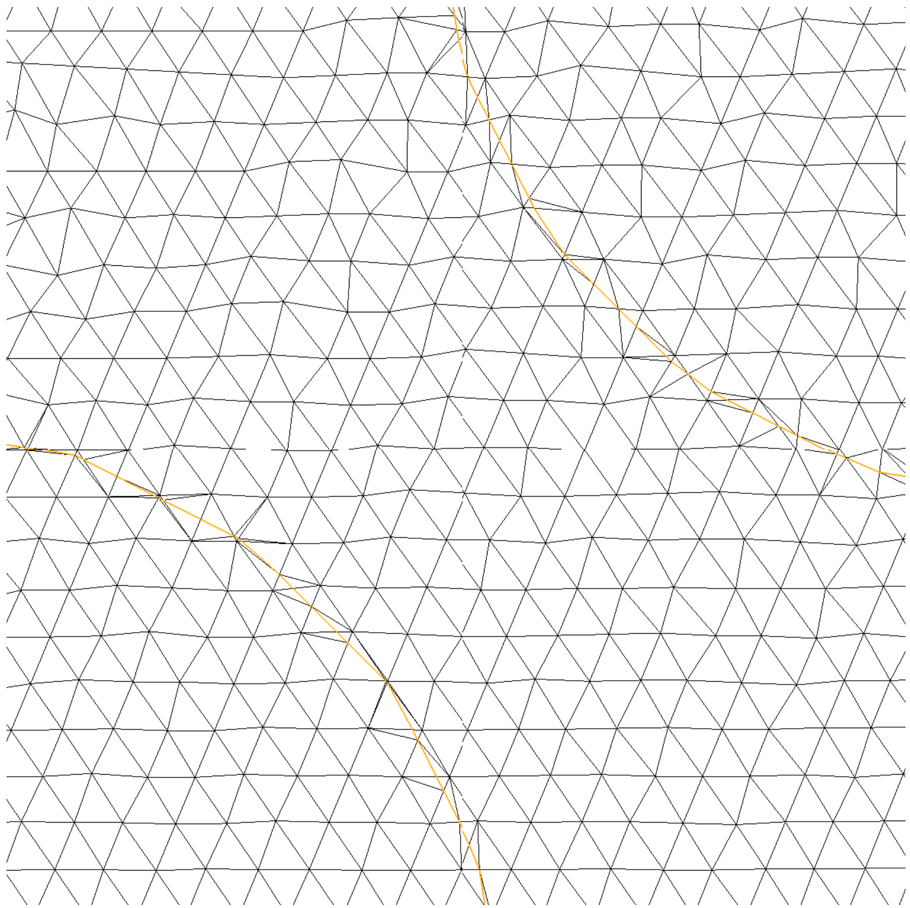


(b) Isolines of its signed distance function

The algorithm in motion

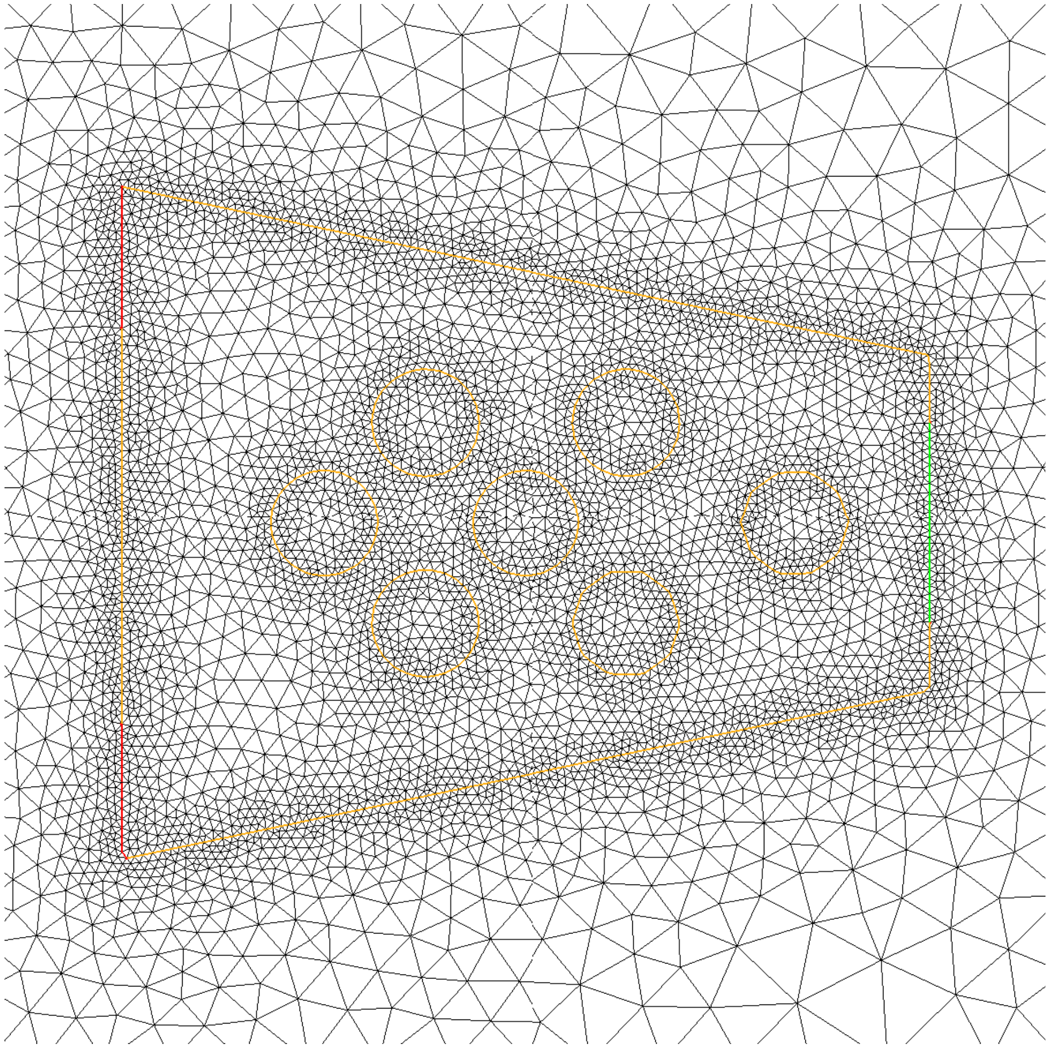
To compute the velocity field through which the shape is to be evolved, a mesh of the volume enclosed by the 0 level set of the distance ($\approx \Omega_0$) is required ; to this end, this 0 isoline is explicitely discretized in the unstructured mesh of $\mathcal{D}$.

Unfortunately, roughly "breaking" this line into the mesh generally yields a very bad-shaped mesh

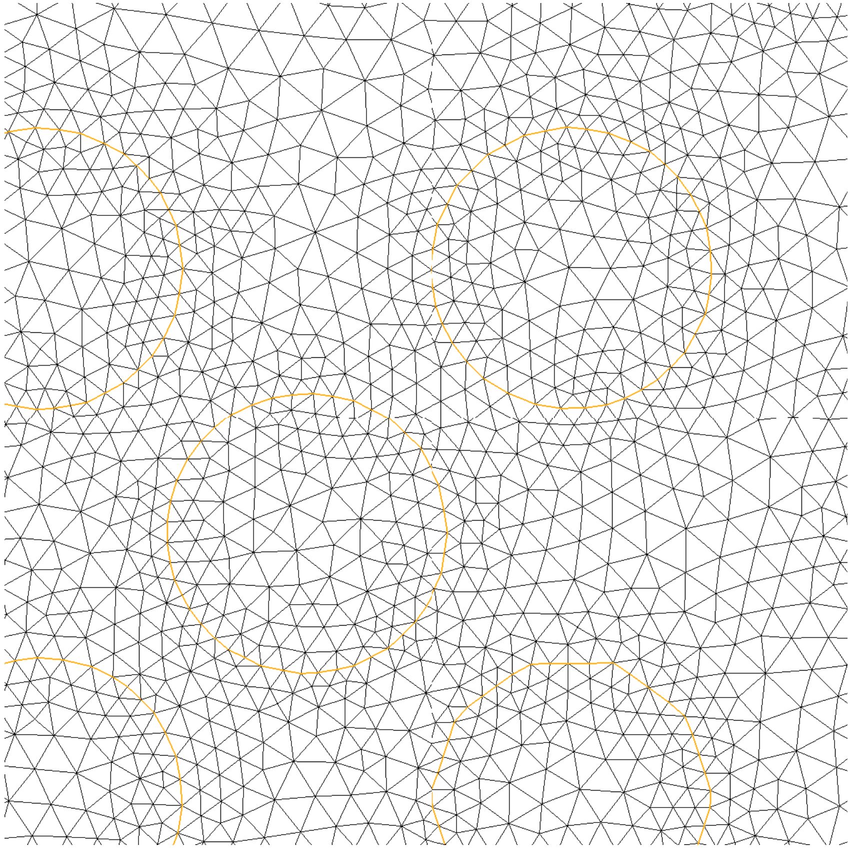


The algorithm in motion

A mesh modification step is then performed, thanks to local mesh operators : close points are collapsed, or added if need be, points are moved to enhance the overall quality of the mesh according to the geometry of the shape.



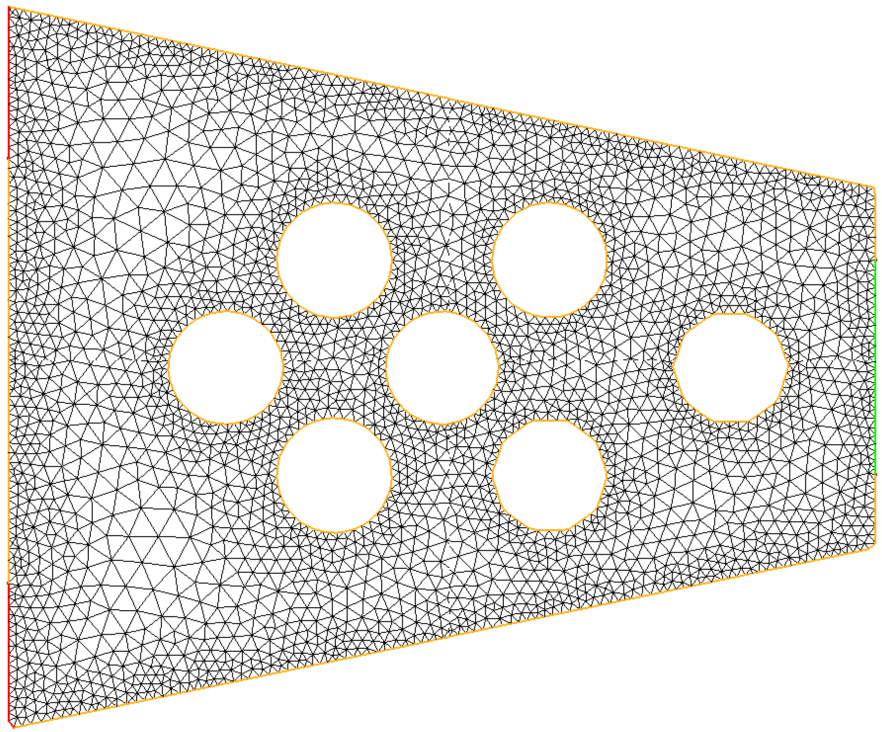
(a) Modified mesh of the computational domain



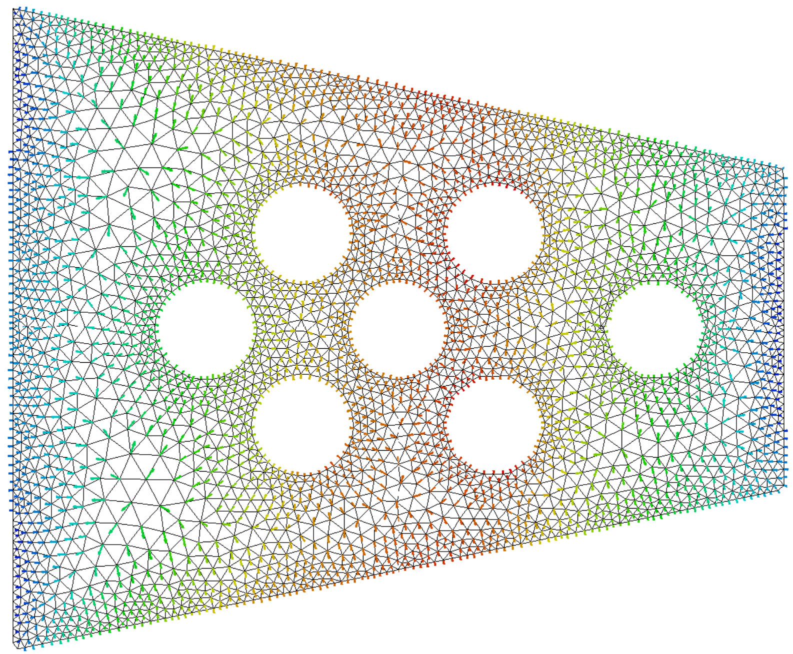
(b) Detail on the mesh

The algorithm in motion

"Forget" the exterior of the shape, and perform the computation of the shape gradient on the shape.



(a) The ‘interior mesh’



(b) Computation of the gradient

The algorithm in motion

"Remember" the computational mesh, and advect the shape as the 0 level set of its signed distance function, computed on the whole computational mesh.

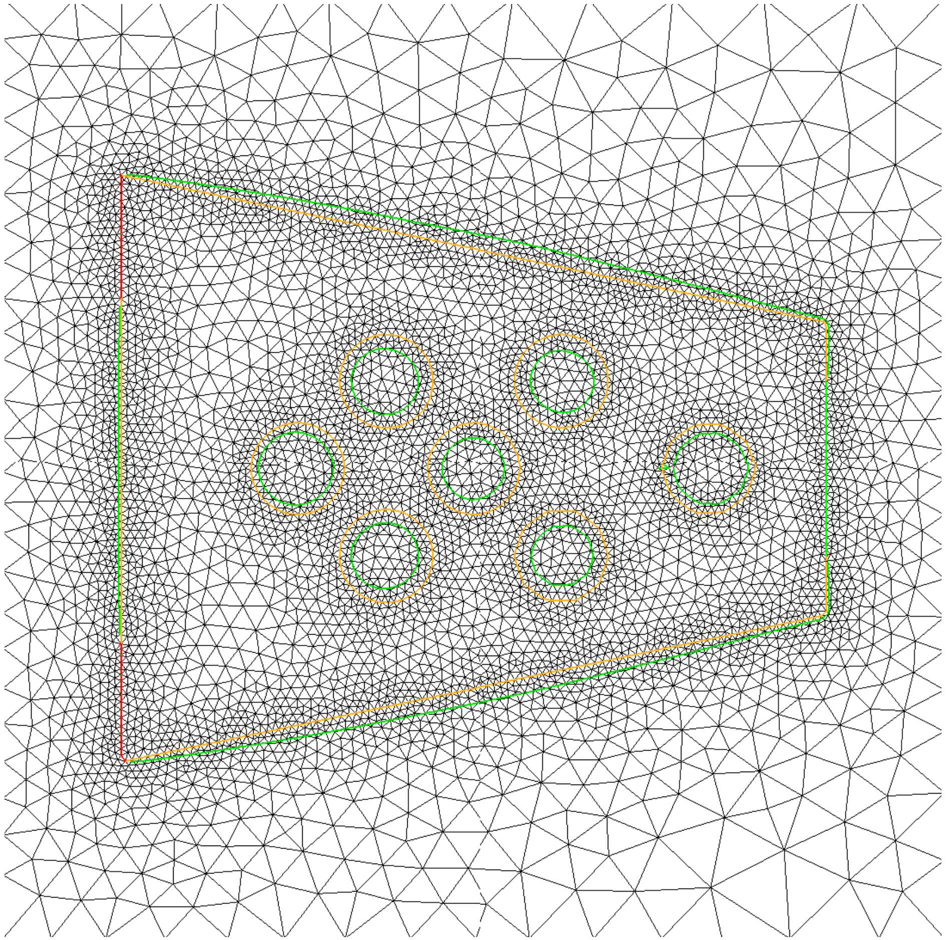
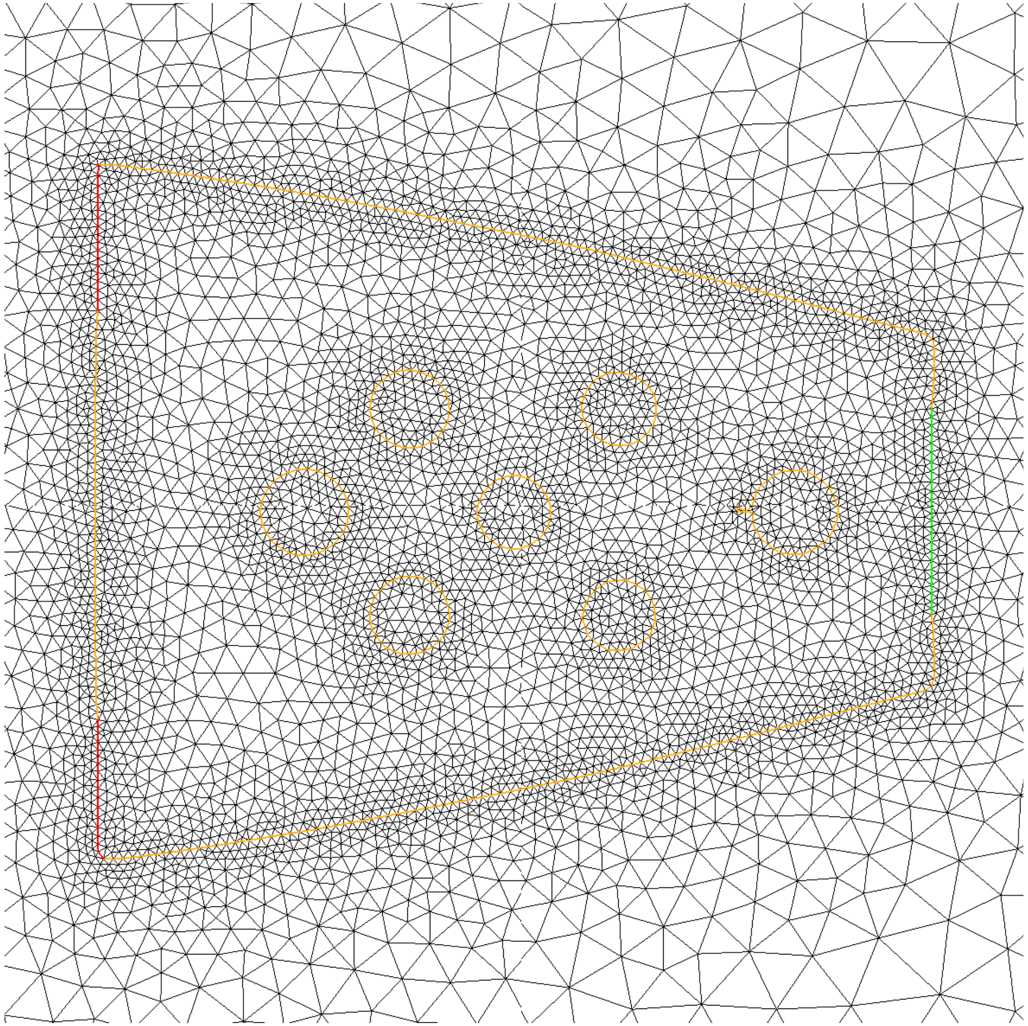


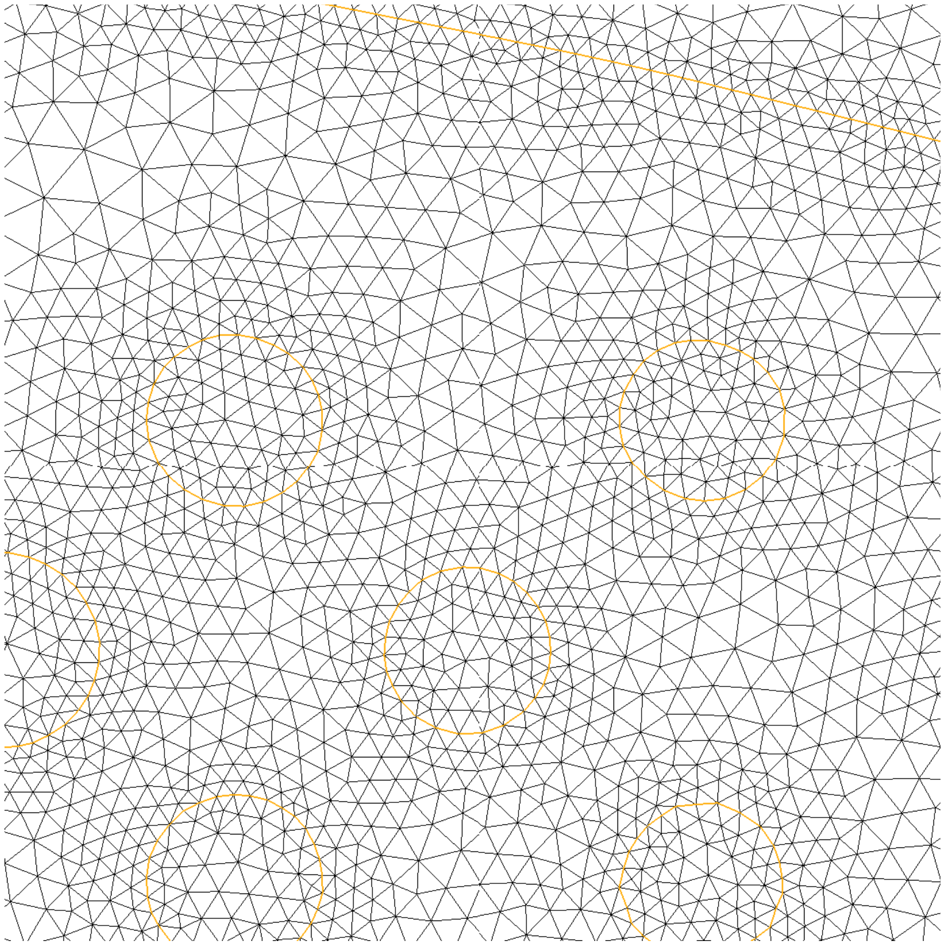
Figure 5: *The previous shape, discretized in the mesh (in yellow), and the 'new', advected* 0*-level set (in green).*

The algorithm in motion

Go on as before, until convergence (discretize the 0 level set in the computational mesh, clean the mesh,...).



(a) The 'interior mesh'

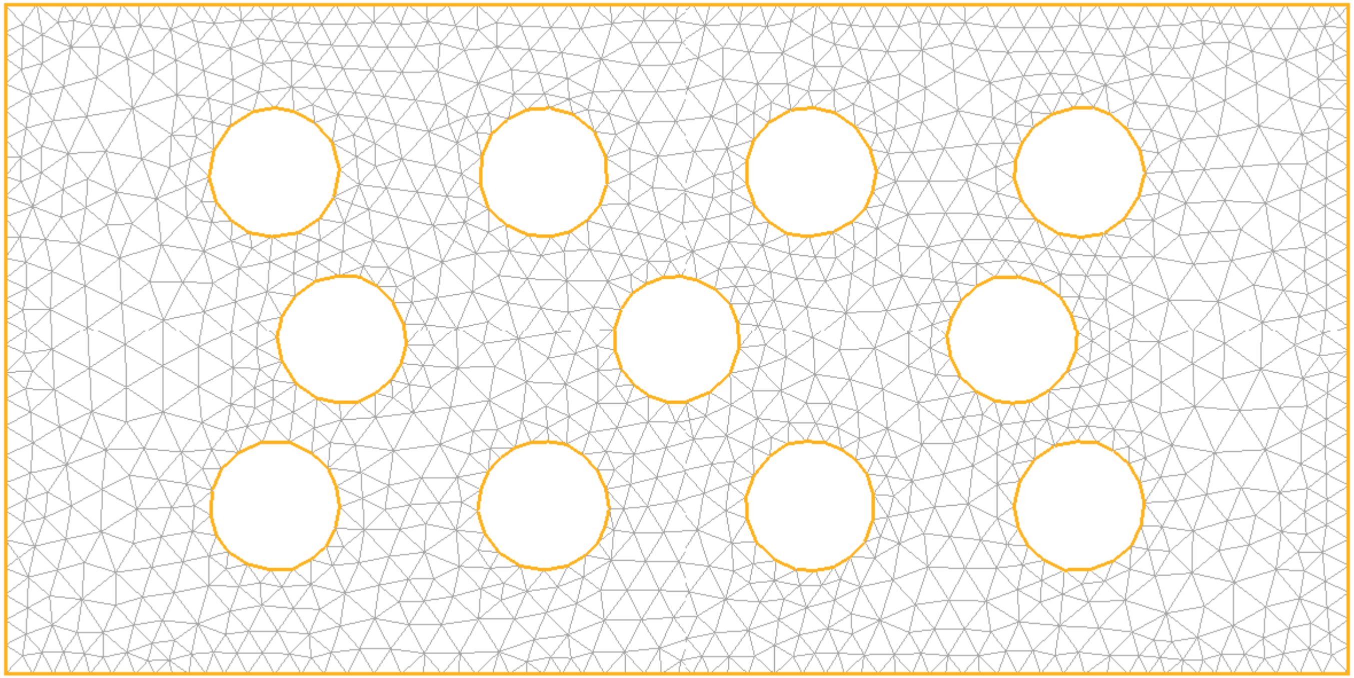


(b) Computation of the gradient

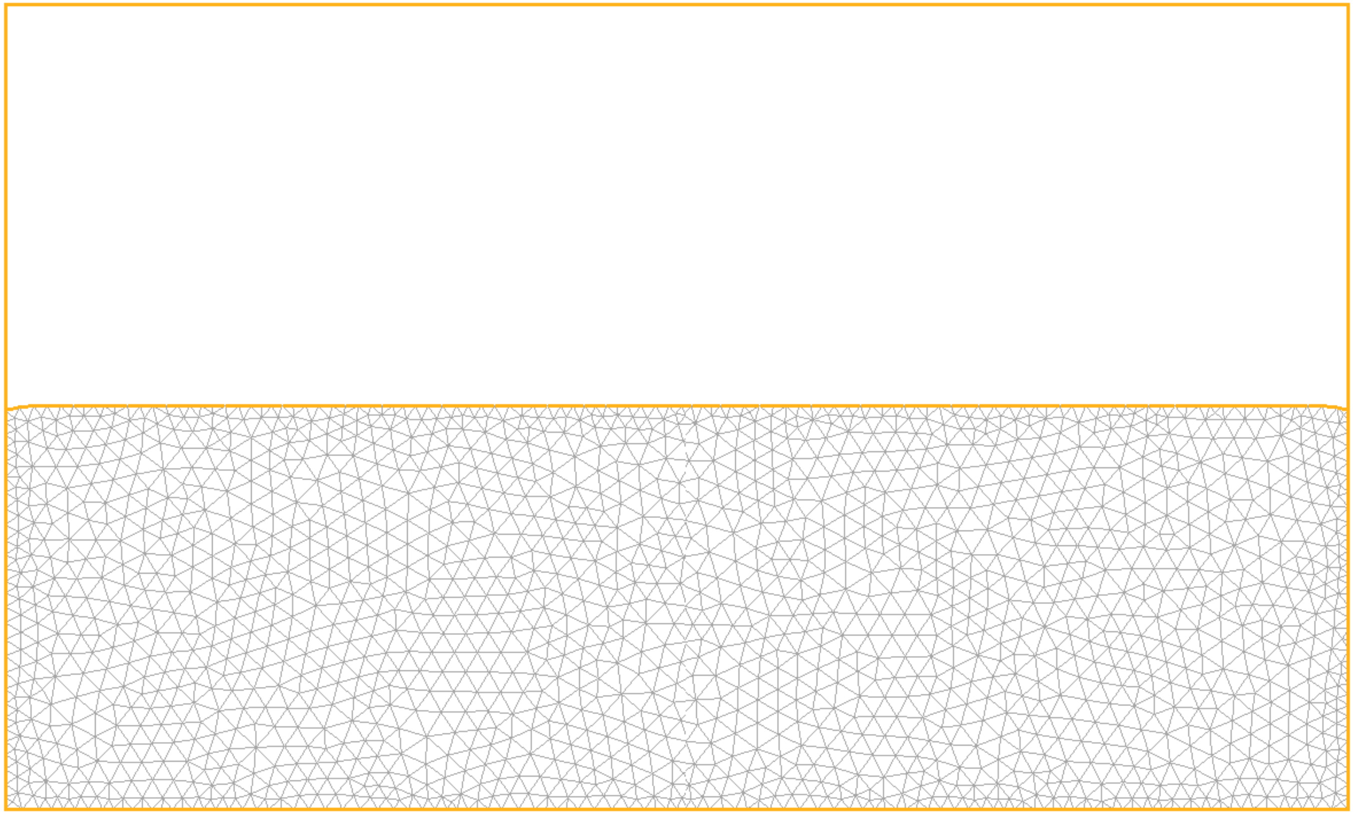
Contents

1. The shape-gradient based method for a model problem in shape optimization
2. The level set method
   - Initializing level set functions
   - Advection of level set functions
3. The proposed algorithm
4. Numerical results

Some numerical results



Some numerical results



Still a long way to go...

- Application to many other shape optimization problems : different cost functionals (quadratic difference to a prescribed displacement, Von Mises stress constraints,...), different mechanical models (elastodynamic,...),...

- Extension of the process to 3d : of course, many technical difficulties are expected, but the whole process has been thought so that such an extension is possible.

Thank you !

Thank you for your attention !