Shape optimization, level set methods on unstructured meshes and mesh evolution

Charles Dapogny[1,2],

under the direction of Marc Albertelli, Grégoire Allaire and Pascal Frey

[1] Laboratoire Jacques-Louis Lions, UPMC, Paris, France

[2] Technocentre Renault, Guyancourt

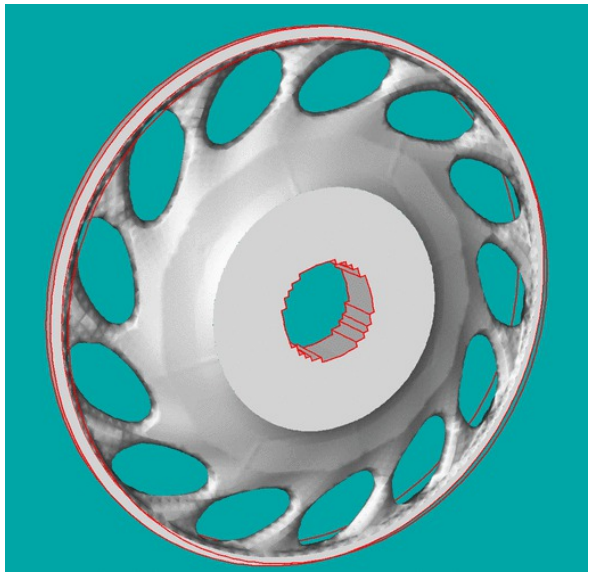
Sketch of the presentation

- **Introduction**
- **A worst-case design approach for shape optimization under uncertainties**
  - The main ideas in an abstract framework
  - Applications in shape optimization:
    - Shape optimization under uncertainties over the body forces
    - Shape optimization under uncertainties over the elastic material
    - Shape optimization under geometric uncertainties
- **Shape optimization by a level set based mesh evolution method**
  - Presentation of the proposed method
  - From a meshed description to a level set description
  - A meshing algorithm for implicit domains
  - The method in action
  - Numerical results
- **Perspectives**

Introduction

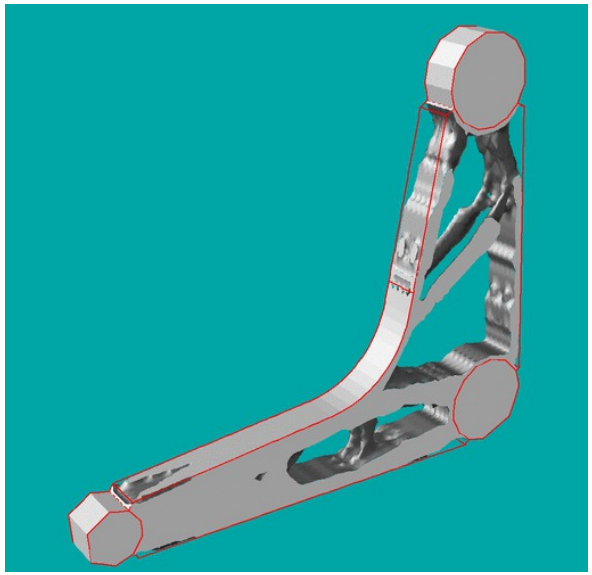
Introduction

The steady increase in the cost of raw materials has made it necessary to optimize mechanical parts from the early stages of design.



Such problems are difficult, partly because

- formulating shape optimization problems in a way which is both realistic and tractable is sometimes difficult.
- they require an accurate description of the various shapes that could be obtained through the optimization process.

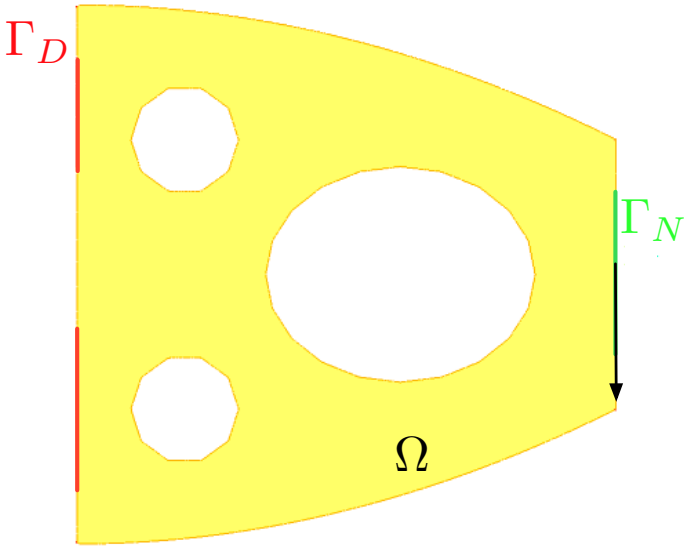


Automatic techniques (implemented in industrial softwares) have started to replace the traditional trial-and-error methods used by engineers, but still leave room for many forthcoming developments.

A model problem in linear elasticity

A shape is a bounded open domain $\Omega \subset \mathbb{R}^d$, which is

- fixed on a part $\Gamma_D \subset \partial\Omega$ of its boundary,
- submitted to surface loads g, applied on $\Gamma_N \subset \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$.



A 'Cantilever' beam

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

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

where $e(u) = \frac{1}{2}(\nabla u^T + \nabla u)$ is the strain tensor field, and A is the Hooke's law of the material.

The deformed cantilever

A model problem in linear elasticity

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

Examples:

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

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

- A least-square discrepancy between the displacement u_Ω and a target displacement $u_0 \in H^1(\Omega)^d$ (useful when designing micro-mechanisms):

$$D(\Omega) = \left(\int_\Omega k(x)||u_\Omega - u_0||^\alpha dx \right)^{\frac{1}{\alpha}},$$

where α is a fixed parameter, and $k(x)$ is a weight factor.

A volume constraint may be enforced with a fixed penalty parameter ℓ:

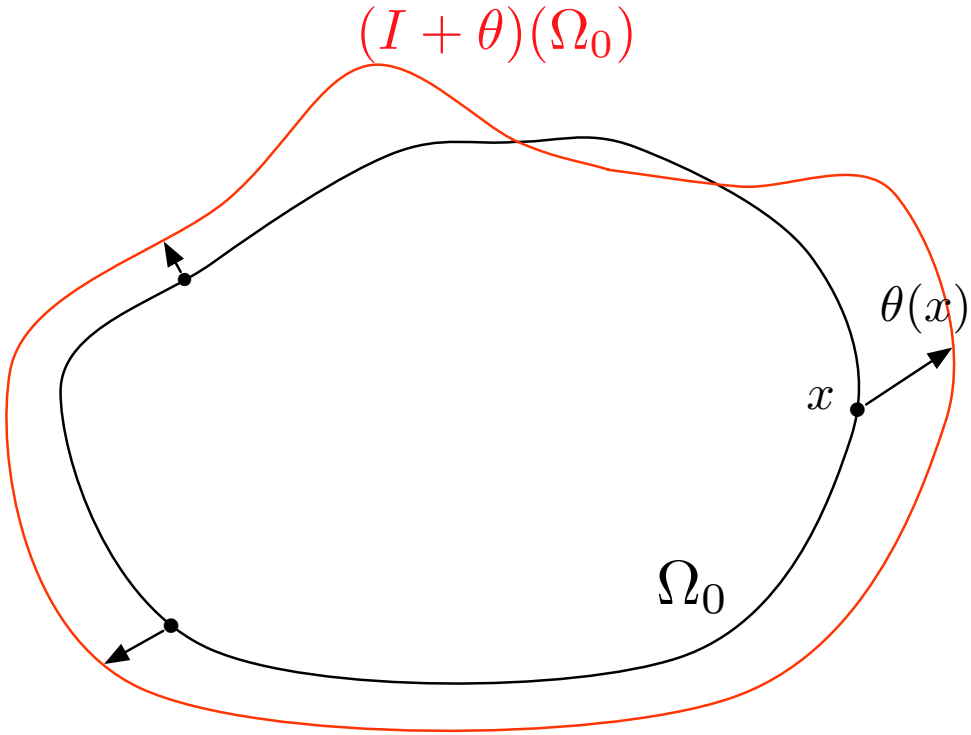
Minimize $J(\Omega) := C(\Omega) + \ell \, \mathrm{Vol}(\Omega)$, or $D(\Omega) + \ell \, \mathrm{Vol}(\Omega)$.

Differentiation with respect to the domain: Hadamard's method

Hadamard's boundary variation method describes variations of a reference, Lipschitz domain Ω_0 of the form:

$$(I+\theta)(\Omega_0),$$

for 'small' $\theta \in W^{1,\infty}\left(\mathbb{R}^d, \mathbb{R}^d\right)$.



DEFINITION 1 *Given a smooth domain Ω_0, a (scalar) function $\Omega \mapsto J(\Omega)$ is shape differentiable at Ω_0 if the function*

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

is Fréchet-differentiable at 0, i.e. the following expansion holds in the vicinity of 0:

$$J((I+\theta)(\Omega_0)) = J(\Omega_0) + J'(\Omega_0)(\theta) + o\left(||\theta||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)}\right).$$

Differentiation with respect to the domain: Hadamard's method

Techniques from optimal control theory make it possible to compute shape gradients; in the case of 'many' functionals of the domain $J(\Omega)$, the shape derivative has the particular structure:

$$J'(\Omega)(\theta) = \int_\Gamma V_\Omega \, \theta \cdot n \, ds,$$

where V_Ω is a scalar field which depends on u_Ω, and possibly on an adjoint state p_Ω.

This shape gradient provides a natural descent direction for J: *for instance*, defining θ as

$$\theta = -V_\Omega n$$

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

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

The generic numerical algorithm

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

1. Compute the solution u_{Ω^n} (and p_{Ω^n}) of the elasticity system on Ω^n.
2. Compute the shape gradient $J'(\Omega^n)$ thanks to the previous 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
- get a mesh of each shape Ω^n so as to perform the required finite element computations.

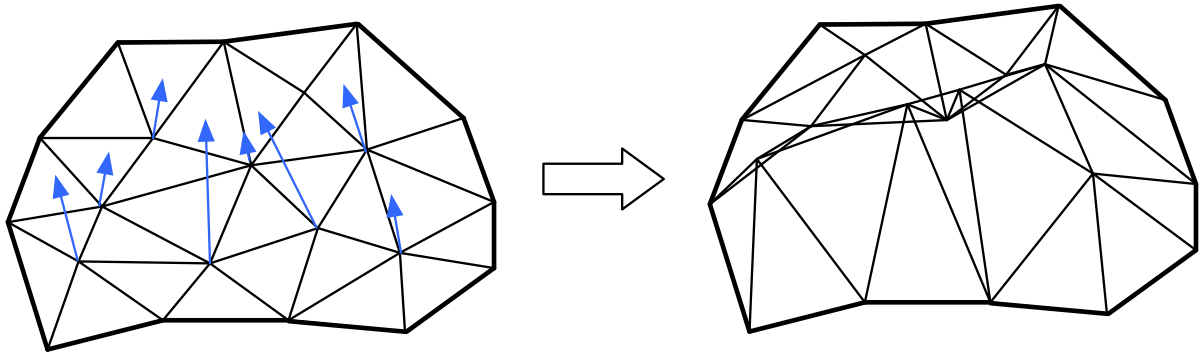


Figure 1: Pushing nodes according to the velocity field may result in an invalid configuration.

A short detour by the Level Set Method

A paradigm: [Osher, Sethian] *the motion of an evolving domain is best described in an implicit way.*

A bounded domain $\Omega \subset \mathbb{R}^d$ is equivalently defined by a function $\phi : \mathbb{R}^d \to \mathbb{R}$ such that:

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

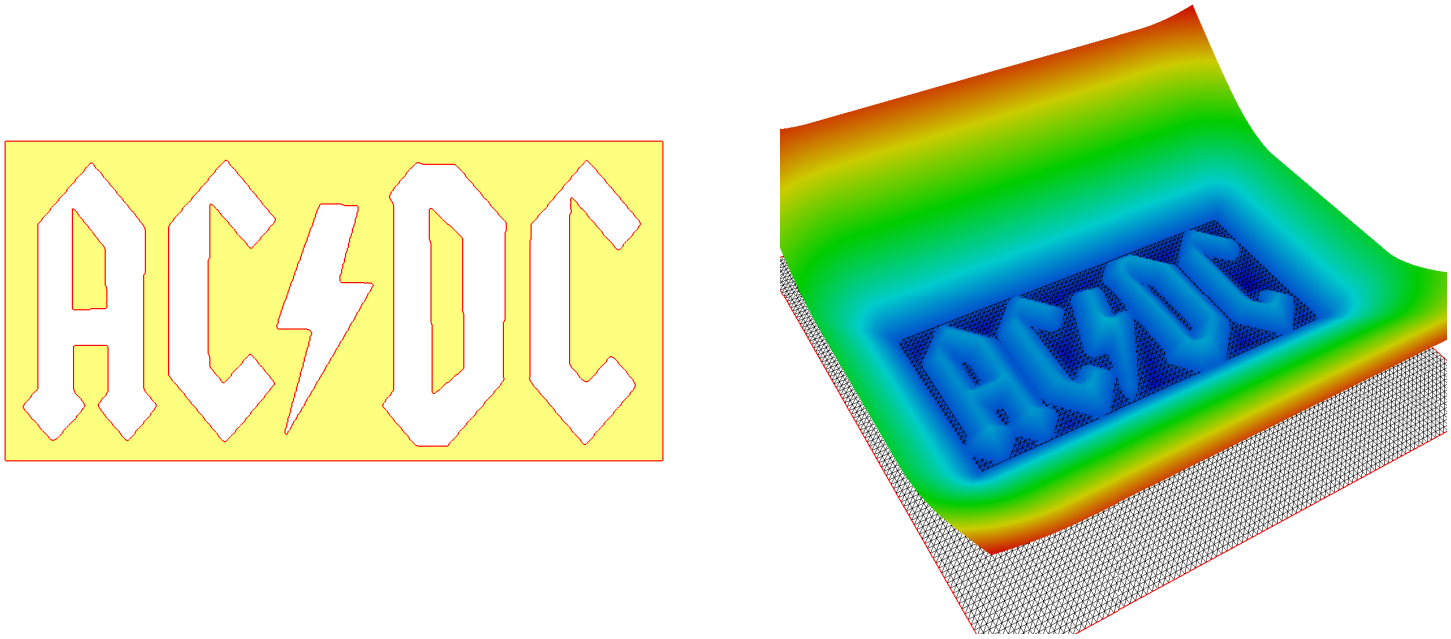


Figure 2: *A bounded domain $\Omega \subset \mathbb{R}^2$ (left), some level sets of an associated level set function (right).*

Surface evolution equations in the level set framework

The motion of an evolving domain $\Omega(t) \subset \mathbb{R}^d$ along a velocity field $v(t,x) \in \mathbb{R}^d$ is translated in terms of an associated 'level set function' $\phi(t,.)$ by the level set advection equation:

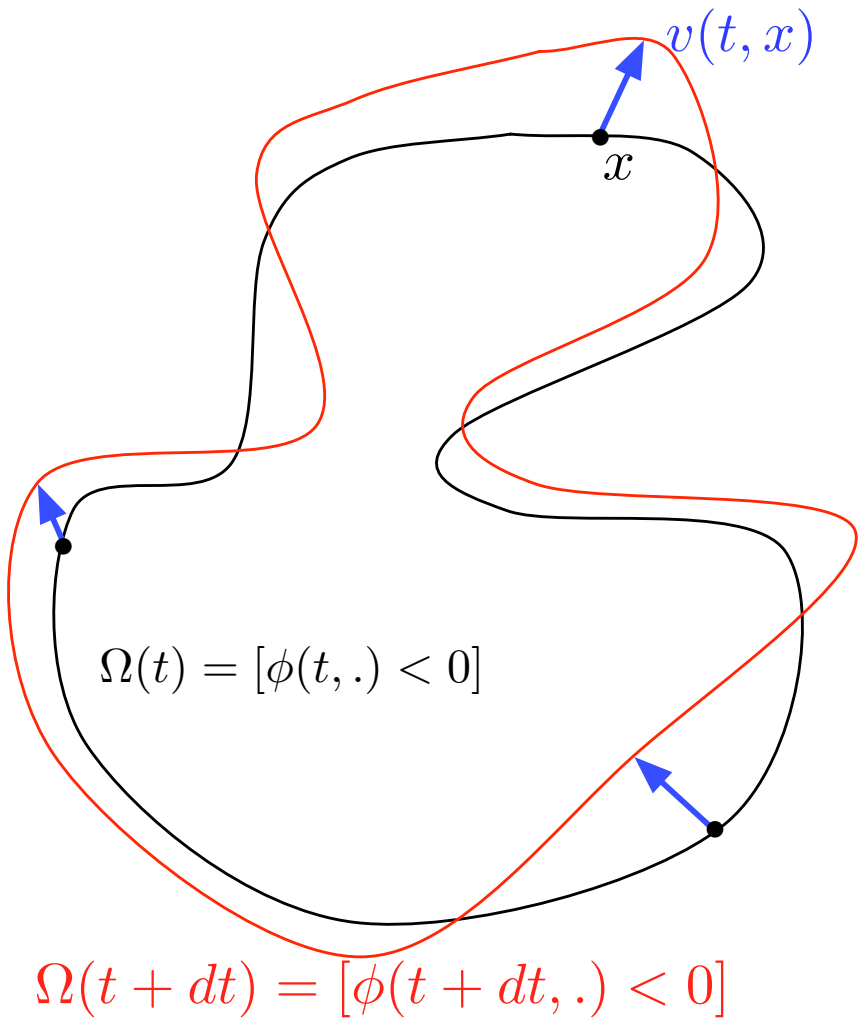
$$\forall t,\ \forall x \in \mathbb{R}^d,\ \frac{\partial \phi}{\partial t}(t,x) + v(t,x).\nabla\phi(t,x) = 0$$

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

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

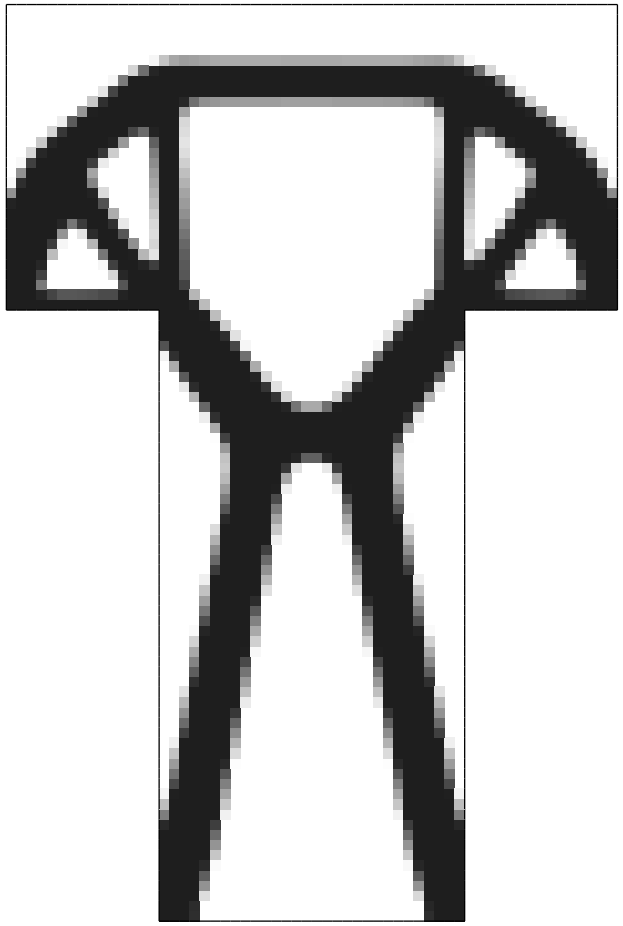
Then the evolution equation rewrites as a Hamilton-Jacobi equation:

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



The level set method for shape optimization of Allaire-Jouve-Toader

- The shapes Ω^n are embedded in a computational box D equipped with a fixed mesh.
- The successive shapes Ω^n are accounted for in the level set framework, i.e. by the knowledge of a function ϕ^n defined on the whole box 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 $D \setminus \Omega^n$ is filled with a very 'soft' material, which leads to an approximate linear elasticity system, defined on D.
- This approach is very versatile and avoids the problem of mesh deformation between iterations.



Shape accounted for with a level set description

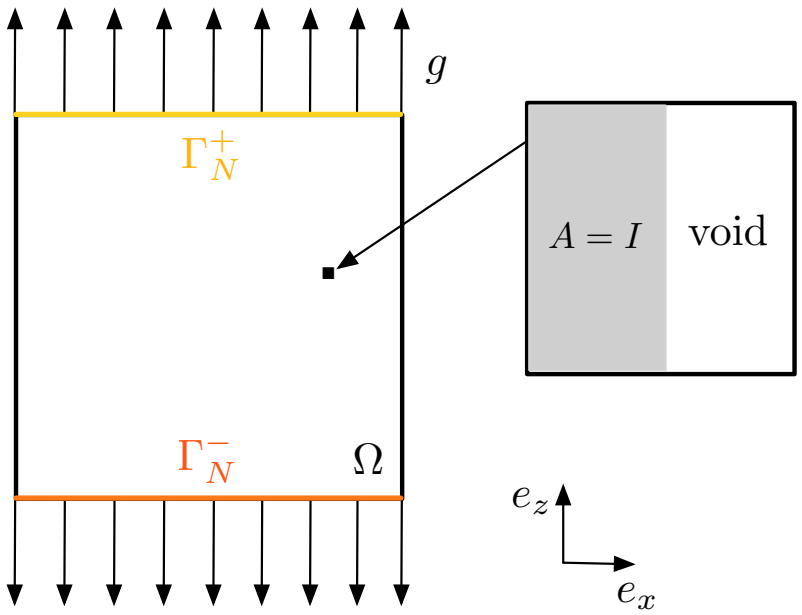
Part I.

Worst-case design in structural optimization

Worst-case design

In realistic models, data may be sullied with unknown perturbations or uncertainties; most often the optimal design strongly depends on these data, whence the need to incorporate a notion of robustness to the model.

Example [Cherkaev]:



- A square domain Ω is subject to surface loads

$$g_0 = \begin{cases} -e_z & \text{on } \Gamma_N^- \\ +e_z & \text{on } \Gamma_N^+ \\ 0 & \text{on } \partial\Omega \setminus (\Gamma_N^+ \cup \Gamma_N^-) \end{cases}.$$

- The optimal microstructure A^* with volume fraction $\theta = 1/2$, for the compliance, is a rank 1 laminate with vertical lamination direction.
- If Ω is now subject to arbitrary surface loads g, the compliance of the resulting structure is:

$$C(A^*) = \begin{cases} 2 & \text{if } g = g_0 \\ \infty & \text{if } g = g_0 + \varepsilon e_x \end{cases}.$$

Worst-case design

When it comes to the optimization of an elastic structure Ω, we may think of:

- Uncertainties over the applied body forces f or surface loads g, due to an inaccurate anticipation of external stresses.

- Uncertainties over the properties of the constituent material of Ω, which may be altered, e.g. because of:
  - variations of conditions in the ambient medium (e.g. temperature, humidity),
  - defects during the manufacturing process (e.g. small inclusions of parasitic material).

- Uncertainties over the geometry of Ω itself; for instance,
  - the manufacturing process of Ω only respects its geometry up to some tolerance,
  - the geometry of Ω may end up perturbed over time, because of wear or erosion.

The main philosophy in an abstract, simplified framework

- $\mathcal{U}_{ad} \subset \mathcal{H}$ is a set of admissible designs h (thickness of plates, geometry of shapes).

- $\mathcal{P}$ is a Banach space of perturbations δ. By convention, $\delta = 0$ stands for the unperturbed problem, and perturbations are assumed to have small amplitude $||\delta||_{\mathcal{P}} \leq m$.

- The physics of the problem are modeled by a state-constrained system:

$$\mathcal{A}(h)u(h,\delta) = b(\delta).$$

  $u(h) := u(h,0)$ is the unperturbed state.

- The cost of a design $h \in \mathcal{U}_{ad}$, perturbed by $\delta \in \mathcal{P}$ is $\mathcal{C}(u(h,\delta))$.

- Minimization of the worst-case objective function when perturbations are applied:

$$\min_{h \in \mathcal{U}_{ad}} \mathcal{J}(h), \quad \mathcal{J}(h) = \max_{||\delta||_{\mathcal{P}} \leq m} \mathcal{C}(u(h,\delta)).$$

The main philosophy in an abstract, simplified framework

Idea: Linearize the cost function with respect to the perturbations δ:

$$\mathcal{C}(u(h,\delta)) \approx \mathcal{C}(u(h)) + \delta \, \frac{d\mathcal{C}}{du}(u(h,0)) \cdot \frac{\partial u}{\partial \delta}(h,0),$$

then consider the approximate worst-case functional:

$$\begin{aligned} \widetilde{\mathcal{J}}(h) &= \mathcal{C}(u(h)) + \max_{||\delta||_{\mathcal{P}} \leq m} \left(\delta \, \tfrac{d\mathcal{C}}{du}(u(h,0)) \cdot \tfrac{\partial u}{\partial \delta}(h,0) \right) \\ &= \mathcal{C}(u(h)) + m \left|\left| \tfrac{d\mathcal{C}}{du}(u(h)) \cdot \tfrac{\partial u}{\partial \delta}(h,0) \right|\right|_{\mathcal{Q}} \end{aligned},$$

where $\mathcal{Q}$ is either the dual (i.e. $\mathcal{Q} = \mathcal{P}^*$) or the pre-dual (i.e. $\mathcal{P} = \mathcal{Q}^*$) Banach space of $\mathcal{P}$.

The main philosophy in an abstract, simplified framework

The (unknown) sensitivity $\frac{\partial u}{\partial \delta}(h,0)$ can be eliminated using an adjoint state $p(h)$.

Differentiating the state equation $\mathcal{A}(h)u(h,\delta) = b(\delta)$ yields:

$$\mathcal{A}(h)\frac{\partial u}{\partial \delta}(h,0) = \frac{db}{d\delta}(0).$$

Thus,
$$\begin{aligned}\frac{d\mathcal{C}}{du}(u(h)) \cdot \frac{\partial u}{\partial \delta}(h,0) &= \mathcal{A}(h)^T p(h) \cdot \frac{\partial u}{\partial \delta}(h,0) \\ &= \left(\mathcal{A}(h)\frac{\partial u}{\partial \delta}(h,0)\right) \cdot p(h) \ , \\ &= \frac{db}{d\delta}(0) \cdot p(h)\end{aligned}$$

where the adjoint state $p(h)$ is defined by: $\mathcal{A}(h)^T p(h) = \frac{d\mathcal{C}}{du}(u(h))$.

The approximate worst-case problem reads:

$$\min_{h \in \mathcal{U}_{ad}} \widetilde{\mathcal{J}}(h), \quad \widetilde{\mathcal{J}}(h) = \underbrace{\mathcal{C}(u(h))}_{\substack{\text{unperturbed} \\ \text{cost function}}} + \underbrace{m \left\| \frac{db}{d\delta}(0) \cdot p(h) \right\|_{\mathcal{Q}}}_{\substack{\text{penalization by} \\ \text{a norm of an} \\ \text{adjoint state}}}.$$

The main philosophy in an abstract, simplified framework

$\widetilde{\mathcal{J}}(h)$ can now be differentiated using 'standard' methods from optimal control theory:

$$\widetilde{\mathcal{J}}(h) = \mathcal{C}(u(h)) + m \left|\left| \frac{db}{d\delta}(0) \cdot p(h) \right|\right|_{\mathcal{Q}}$$

$$\widetilde{\mathcal{J}}'(h)(s) = \underbrace{F(u(h), p(h))(s)}_{\text{classical differentiation of } \mathcal{C}(u(h))} + m\, G(u(h), p(h), q(h), z(h))(s)$$

$q(h)$: second adjoint state, for the derivation of $||.||_{\mathcal{Q}}$

$z(h)$: third adjoint state, for the dependence of $p(h)$ on h through $u(h)$

Remark: The above approach is formal; yet, in some particular cases of importance where $\mathcal{J}(h)$ can be computed explicitly, one can prove that:

$$\mathcal{J}'(h)(s) = \widetilde{\mathcal{J}}'(h)(s) + \mathcal{O}(m^2).$$

Shape optimization under uncertainties over the body forces

- Set of admissible shapes: $\mathcal{H} = \mathcal{U}_{ad} = \left\{\Omega \subset \mathbb{R}^d \text{ open and bounded, } \Gamma_D \cup \Gamma_N \subset \partial\Omega\right\}$.
- Variations of admissible shapes: $\Theta_{ad} = \left\{\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d),\ \theta = 0 \text{ on } \Gamma_D \cup \Gamma_N\right\}$.
- The displacement $u_{\Omega, f+\xi}$ of a shape Ω under perturbations $\xi \in \mathcal{P} := L^2(\mathbb{R}^d)^d$ over the body forces reads:

$$\left\{\begin{array}{rcll} -\text{div}(Ae(u)) & = & f+\xi & \text{in } \Omega \\ u & = & 0 & \text{on } \Gamma_D \\ Ae(u)n & = & g & \text{on } \Gamma_N \\ Ae(u)n & = & 0 & \text{on } \Gamma \end{array}\right.$$

- Optimization of the worst-case compliance:

$$\mathcal{J}(\Omega) = \sup_{\substack{\xi \in L^2(\mathbb{R}^d)^d \\ ||\xi||_{L^2(\mathbb{R}^d)^d} \leq m}} \mathcal{C}(\Omega, f+\xi),\ \mathcal{C}(\Omega, f) = \int_\Omega f \cdot u_{\Omega,f}\, dx + \int_{\Gamma_N} g \cdot u_{\Omega,f}\, ds,$$

  approximated by

$$\widetilde{\mathcal{J}}(\Omega) = \mathcal{C}(\Omega, f) + 2m\, ||u_\Omega||_{L^2(\Omega)^d}.$$

- Other approaches exist for this very specific problem [de Gournay, Allaire, Jouve], [Cherkaev], which are more accurate and not restricted to *small* perturbations.

Shape optimization under uncertainties over the body forces

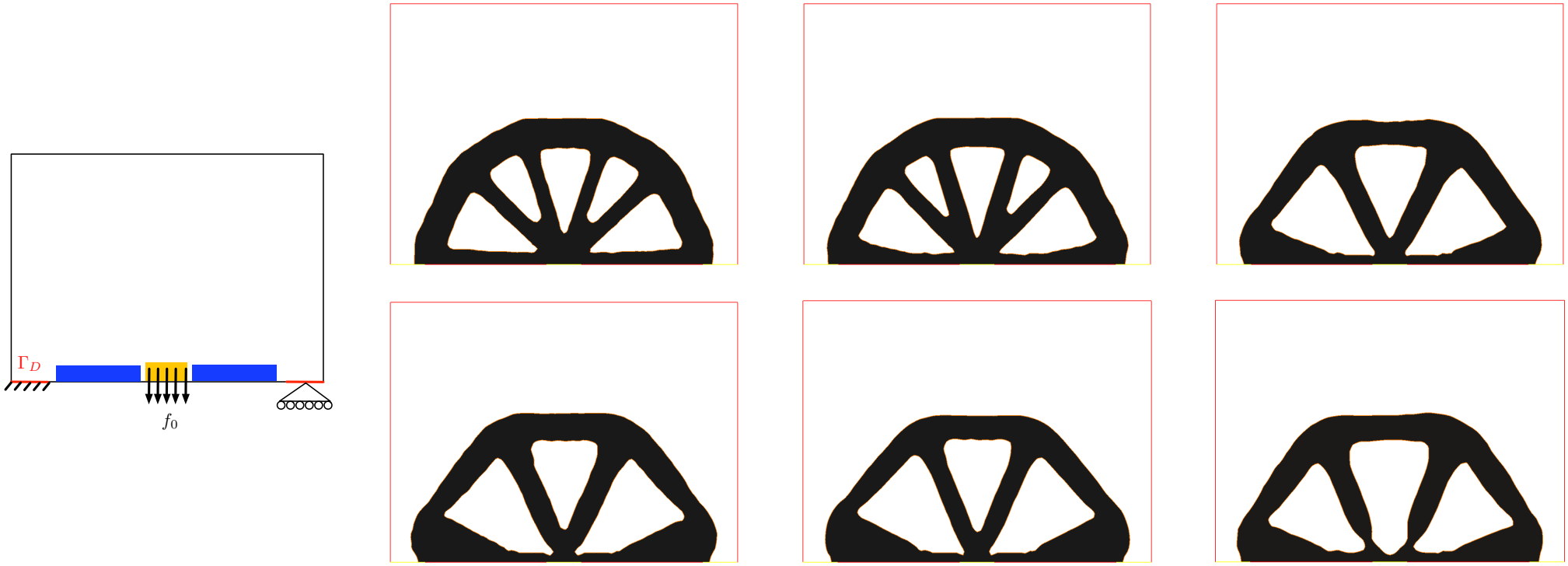


Figure 3: *(From left to right, top to bottom):* optimal shape for the worst-case optimal bridge example, for $m = 0$, $0.2, 0.5, 1, 1.5, 2$. A volume constraint $V = V_T = 0.75$ is imposed in all six cases.

Shape optimization under uncertainties over the elastic material

- The displacement $u_{\Omega,\lambda+\alpha,\mu+\beta}$ of Ω under perturbations $(\alpha,\beta) \in \mathcal{P} := L^\infty(\mathbb{R}^d)^2$ over the Lamé coefficients of the elastic material is the solution to:

$$\left\{\begin{array}{rcll} -\mathrm{div}(A_{\lambda+\alpha,\mu+\beta}e(u)) & = & f & \text{in } \Omega \\ u & = & 0 & \text{on } \Gamma_D \\ A_{\lambda+\alpha,\mu+\beta}e(u)n & = & g & \text{on } \Gamma_N \\ A_{\lambda+\alpha,\mu+\beta}e(u)n & = & 0 & \text{on } \Gamma \end{array}\right. , \quad A_{\lambda,\mu}e = 2\mu e + \lambda \mathrm{tr}(e) I.$$

- Optimization of the worst-case least-square criterion:

$$\mathcal{J}(\Omega) = \sup_{\substack{(\alpha,\beta)\in L^\infty(\mathbb{R}^d)^2 \\ ||(\alpha,\beta)||_{L^\infty(\mathbb{R}^d)^2}\leq m}} \mathcal{C}(\Omega,\lambda+\alpha,\mu+\beta), \ \mathcal{C}(\Omega,\lambda,\mu) = \int_\Omega j(u_{\Omega,\lambda,\mu})\,dx,$$

approximated by:

$$\widetilde{\mathcal{J}}(\Omega) = \mathcal{C}(\Omega,\lambda,\mu) + 2m\,||e(u_\Omega) : e(p_\Omega)||_{L^1(\Omega)} + m\,||\mathrm{div}(u_\Omega)\mathrm{div}(p_\Omega)||_{L^1(\Omega)},$$

where the adjoint state p_Ω solves:

$$\left\{\begin{array}{rcll} -\mathrm{div}(Ae(p)) & = & -j'(u_\Omega) & \text{in } \Omega \\ p & = & 0 & \text{on } \Gamma_D \\ Ae(p)n & = & 0 & \text{on } \Gamma \cup \Gamma_N \end{array}\right. .$$

Shape optimization under uncertainties over the elastic material

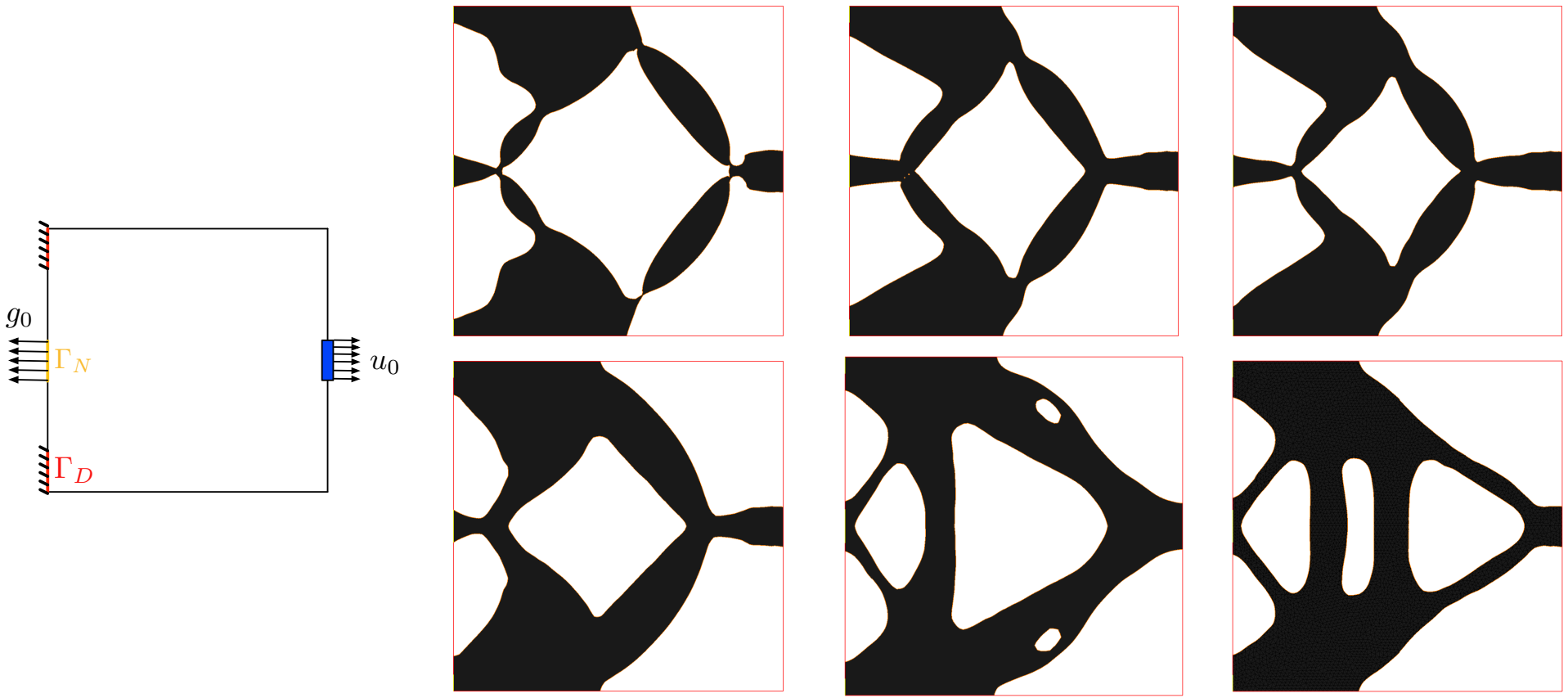


Figure 4: *(From left to right, top to bottom):* optimal shape for the worst-case force inverter test case ($j(u) = k(x)|u - u_0|^2$), under perturbations over the Lamé coefficients of the material of magnitude $m = 0, 0.002, 0.003, 0.0075, 0.02, 0.1$.

Shape optimization under geometric uncertainties

Optimization of a worst-case penalization of stress:

$$\mathcal{J}(\Omega) = \sup_{\substack{V \in W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d) \\ ||V||_{L^\infty(\mathbb{R}^d)^d} \leq m}} \mathcal{C}((I+\chi V)(\Omega)), \;\; \mathcal{C}(\Omega) = \int_\Omega j(\sigma(u_\Omega))\, dx,$$

approximated by:

$$\widetilde{\mathcal{J}}(\Omega) = \int_\Omega j(\sigma(u_\Omega))\, ds + m \int_\Gamma \chi \; |j(\sigma(u_\Omega)) + Ae(u_\Omega) : e(p_\Omega) - f \cdot p_\Omega| \; ds,$$

where p_Ω is an adjoint state.

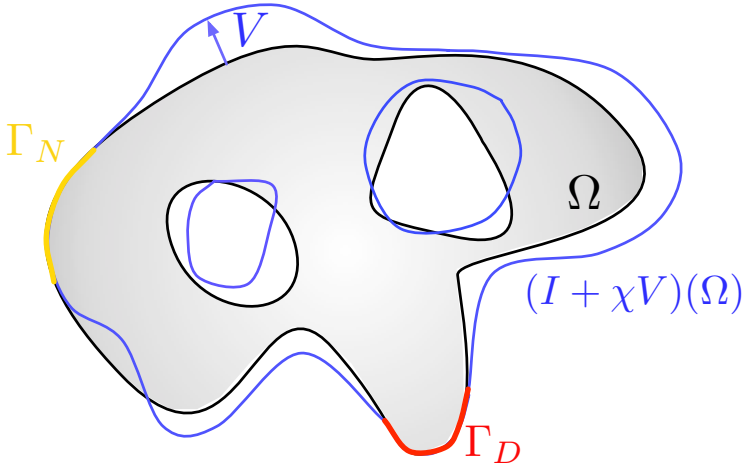


Figure 5: Perturbation of the free boundary Γ of a shape Ω by a small vector field V.

Shape optimization under geometric uncertainties

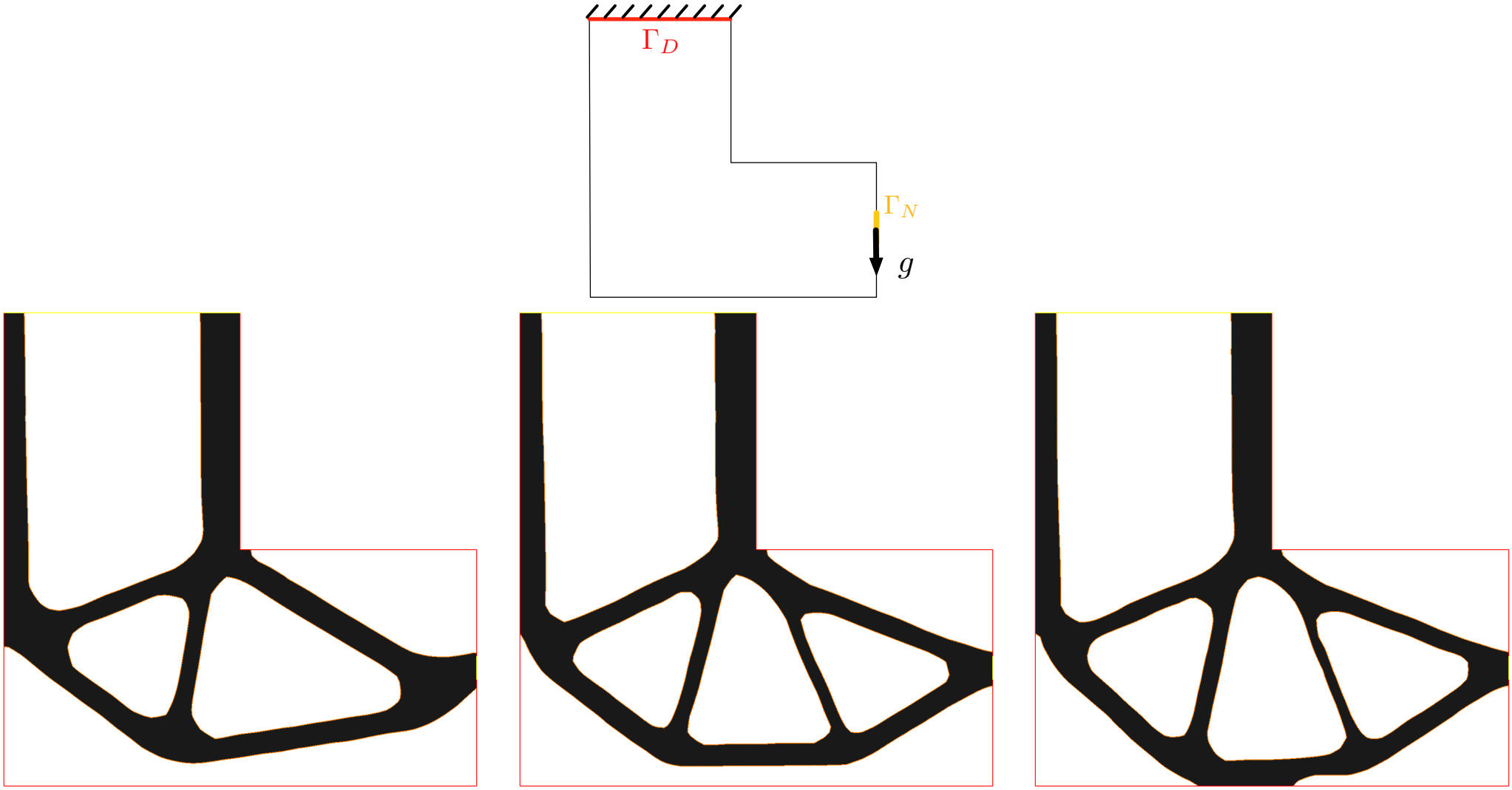


Figure 6: *(From left to right):* optimal shape for $m = 0, 0.01, 0.02$, for the L-Beam example: $j(\sigma) = ||\sigma||^5$. A volume constraint $V = V_T = 0.8$ is imposed in all three cases.

Part II.

A mesh evolution method for geometric shape optimization

- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

Part II.

A mesh evolution method for geometric shape optimization

- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

The proposed method for handling mesh evolution

The mesh $\mathcal{T}^n$ of the computational box D is unstructured and changes at each iteration n, so that Ω^n is explicitly discretized in $\mathcal{T}^n$.

- Finite element analyses are performed on Ω^n by 'forgetting' the part of $\mathcal{T}^n$ for the void $D \setminus \Omega^n$.
- The advection step $\Omega^n \rightarrow \Omega^{n+1}$ is held on the whole $\mathcal{T}^n$, using a level set description ϕ^n of Ω^n.

Computation of a descent direction θ^n

$$(\mathcal{T}^n, \Omega^n) \dashrightarrow ? \dashrightarrow (\mathcal{T}^{n+1}, \Omega^{n+1})$$

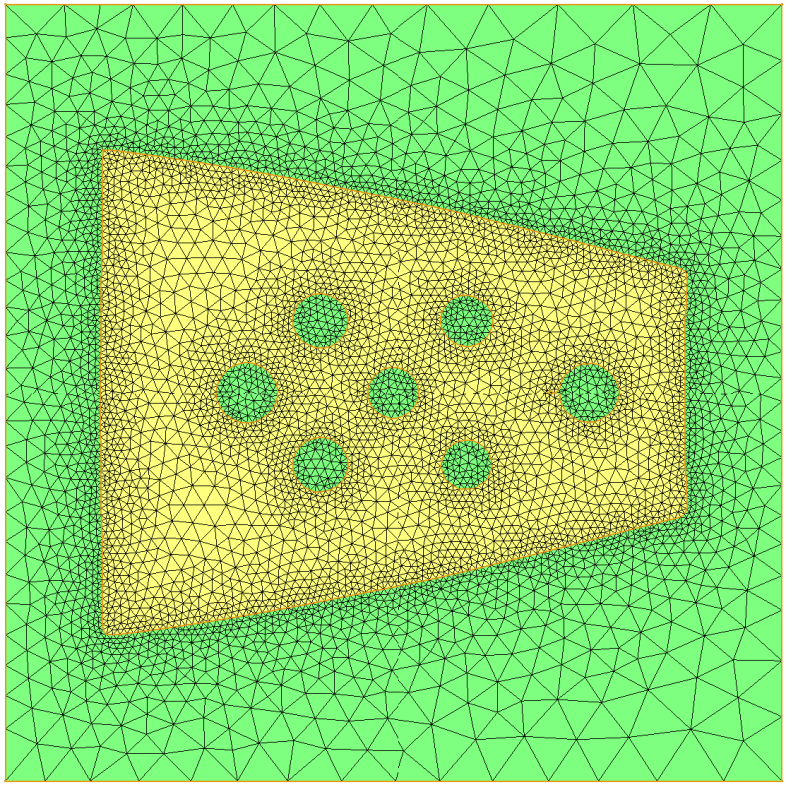
Generation of a level set function on an unstructured mesh

Explicit discretization of an implicit domain in the ambient mesh

$$(\mathcal{T}^n, \phi^n) \longrightarrow (\mathcal{T}^n, \phi^{n+1})$$

Resolution of the advection equation on $(0, \tau^n) \times D$:

$$\begin{cases} \frac{\partial \phi}{\partial t} + \theta^n \cdot \nabla \phi = 0 \\ \phi(t = 0, .) = \phi^n. \end{cases}$$



Shape equipped with a mesh, conformally embedded in a mesh of the computational box.

Part II.

A mesh evolution method for geometric shape optimization

- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

Initializing level-set functions with the signed distance function

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

$$d_\Omega(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega \\ 0 & \text{if } x \in \partial\Omega \\ d(x, \partial\Omega) & \text{if } x \in \overline{{}^c\Omega} \end{cases}$$ *, 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, mainly owing to its unit gradient property:

$$||\nabla d_\Omega(x)|| = 1, \quad \text{p.p } x \in \mathbb{R}^d.$$

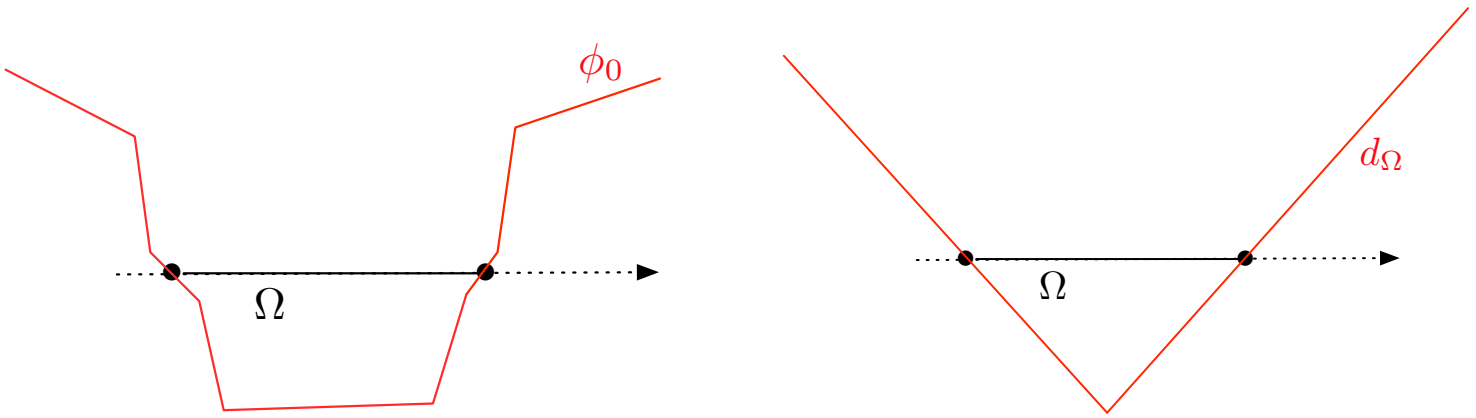


Figure 7: (left) Any level set function for $\Omega = (0, 1) \subset \mathbb{R}$; (right) signed distance function to Ω.

The signed distance function as the steady state of a PDE

- Many existing approaches: Fast Marching Method [Sethian], Fast Sweeping method [Zhao], mostly on Cartesian grids, or particular unstructured meshes.
- Another point of view [Chopp]: suppose $\Omega \subset \mathbb{R}^d$ is implicitly known as

$$\Omega = \left\{ x \in \mathbb{R}^d ; \phi_0(x) < 0 \right\} \text{ and } \partial\Omega = \left\{ x \in \mathbb{R}^d ; \phi_0(x) = 0 \right\},$$

where ϕ_0 is a function we only suppose continuous. Then the function d_Ω can be considered as the steady state of the so-called unsteady Eikonal equation

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \mathrm{sgn}(\phi_0)(||\nabla \phi|| - 1) = 0 & \forall t > 0, x \in \mathbb{R}^d \\ \phi(t = 0, x) = \phi_0(x) & \forall x \in \mathbb{R}^d \end{cases}. \tag{1}$$

THEOREM 1 *[Aubert, Aujol] Define function* ϕ, $\forall x \in \mathbb{R}^d$, $\forall t \in \mathbb{R}_+$,

$$\phi(t,x) = \begin{cases} \mathrm{sgn}(\phi_0(x)) \inf\limits_{||y|| \leq t} (\mathrm{sgn}(\phi_0(x))\phi_0(x+y) + t) & \text{if } t \leq d(x, \partial\Omega) \\ \mathrm{sgn}(\phi_0(x)) d(x, \partial\Omega) & \text{if } t > d(x, \partial\Omega) \end{cases} \tag{2}$$

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

The signed distance function as the steady state of a PDE

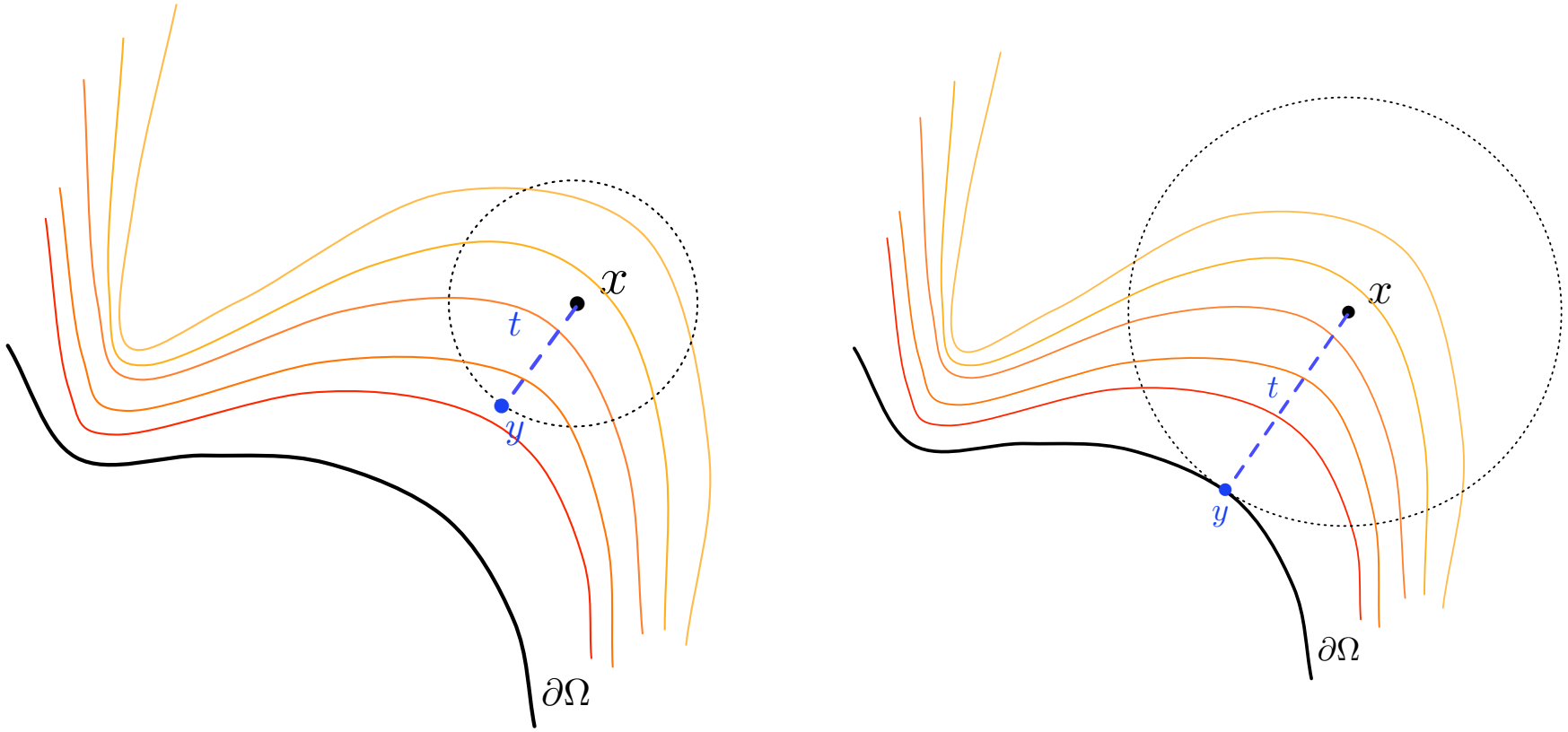


Figure 8: Some level sets of function ϕ_0; *(left):* computation of $\phi(t,x) = \phi_0(y) + t$ for small t; *(right):* computation of $\phi(t,x) = \phi_0(y) + t = d(x,\partial\Omega)$ at $t = d(x,\partial\Omega)$.

The proposed algorithm

Basic idea: *Compute iteratively the solution $\phi(t,x)$, using the exact formula.*

Let dt be a time step, and $t^n = ndt$.

The continuous formula for ϕ can be made iterative: denoting $\phi^n(x) = \phi(t^n, x)$, we have, for $n = 0, \ldots$

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

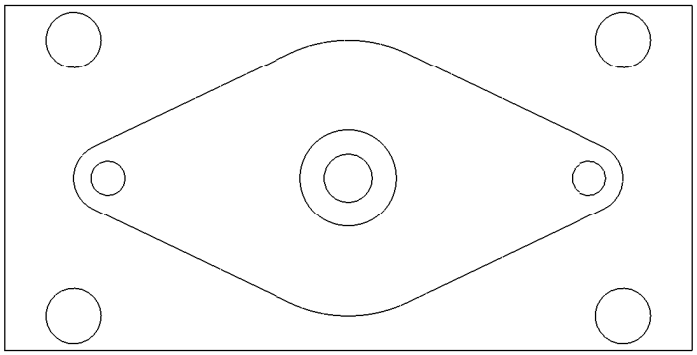
$$\forall x \in \Omega,\ \phi^{n+1}(x) = \sup_{||y|| \leq dt} \phi^n(x+y) - dt$$

and, dt being small enough, the above infimum and supremum are evaluated by taking y in the gradient direction; at a vertex x of the computational mesh $\mathcal{T}$:

$$\forall x \in {}^c\Omega,\ \phi^{n+1}(x) \approx \inf_{T \in \mathrm{Ball}(x)} \phi^n\left(x - dt\frac{\nabla\phi^n|_T}{||\nabla\phi_n|_T||}\right) + dt$$

$$\forall x \in \Omega,\ \phi^{n+1}(x) \approx \sup_{T \in \mathrm{Ball}(x)} \phi^n\left(x + dt\frac{\nabla\phi^n|_T}{||\nabla\phi_n|_T||}\right) - dt.$$

A 2d computational example



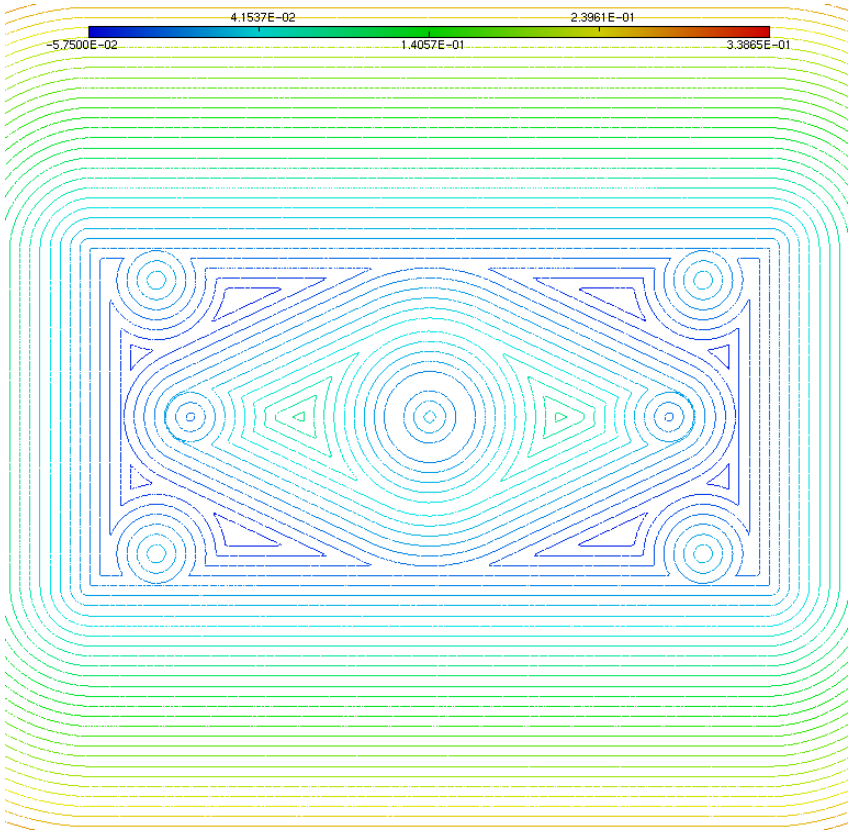


Figure 9: *Computation of the signed distance function to a discrete contour (left), on a fine background mesh (≈ 250000 vertices).*

A 3d example... the 'Aphrodite'.

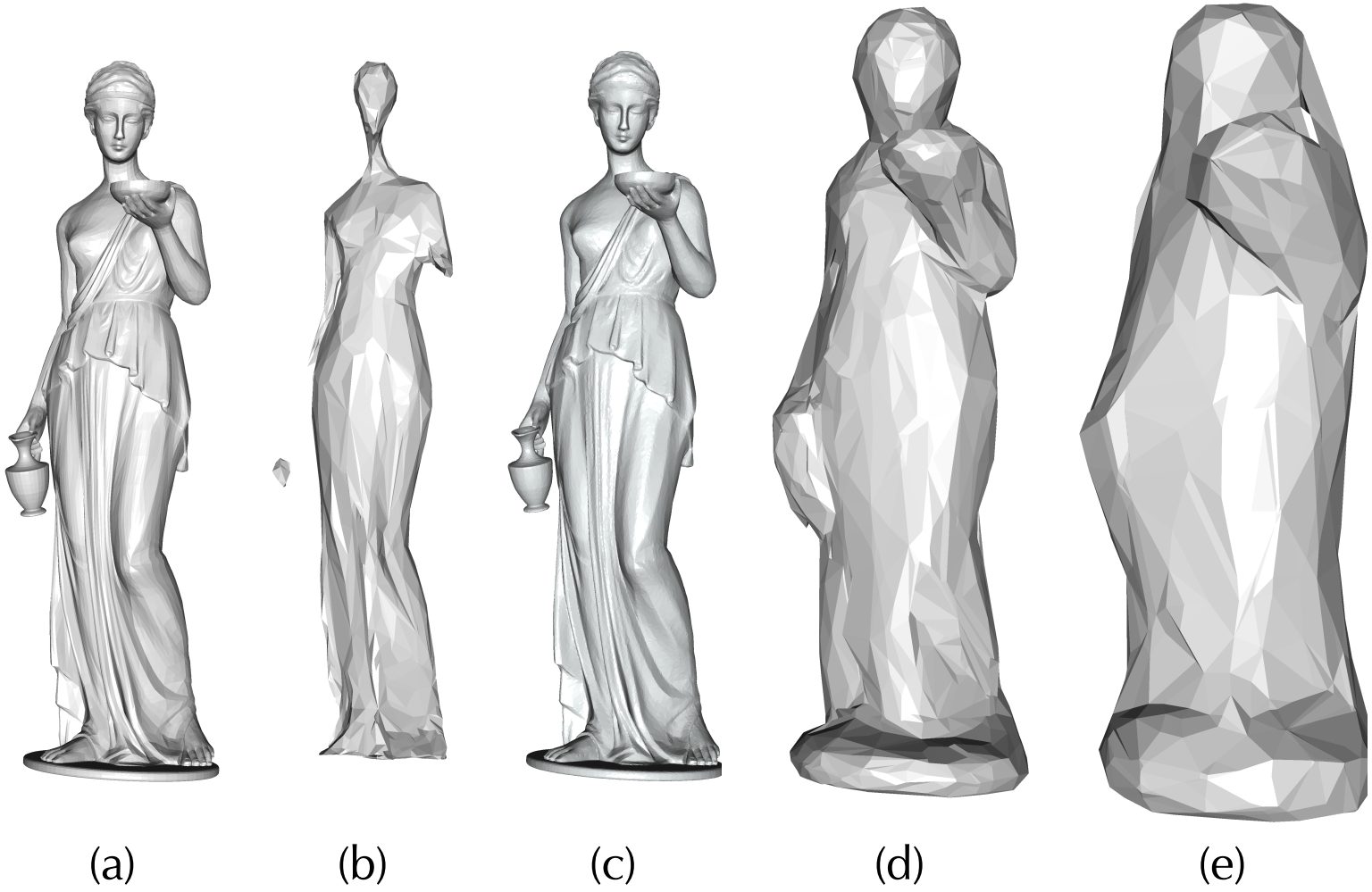


Figure 10: *Isosurfaces of the signed distance function to the 'Aphrodite' (a): (b): isosurface* -0.01*, (c): isosurface* 0*, (d): isosurface* 0.02*, (e): isosurface* 0.05*.*

Part II.

A mesh evolution method for geometric shape optimization

- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

Meshing the negative subdomain of a level set function

Discretizing explicitely the 0 level set of a scalar function ϕ defined at the vertices of a simplicial mesh $\mathcal{T}$ of a computational box D is relatively easy, resorting to patterns.

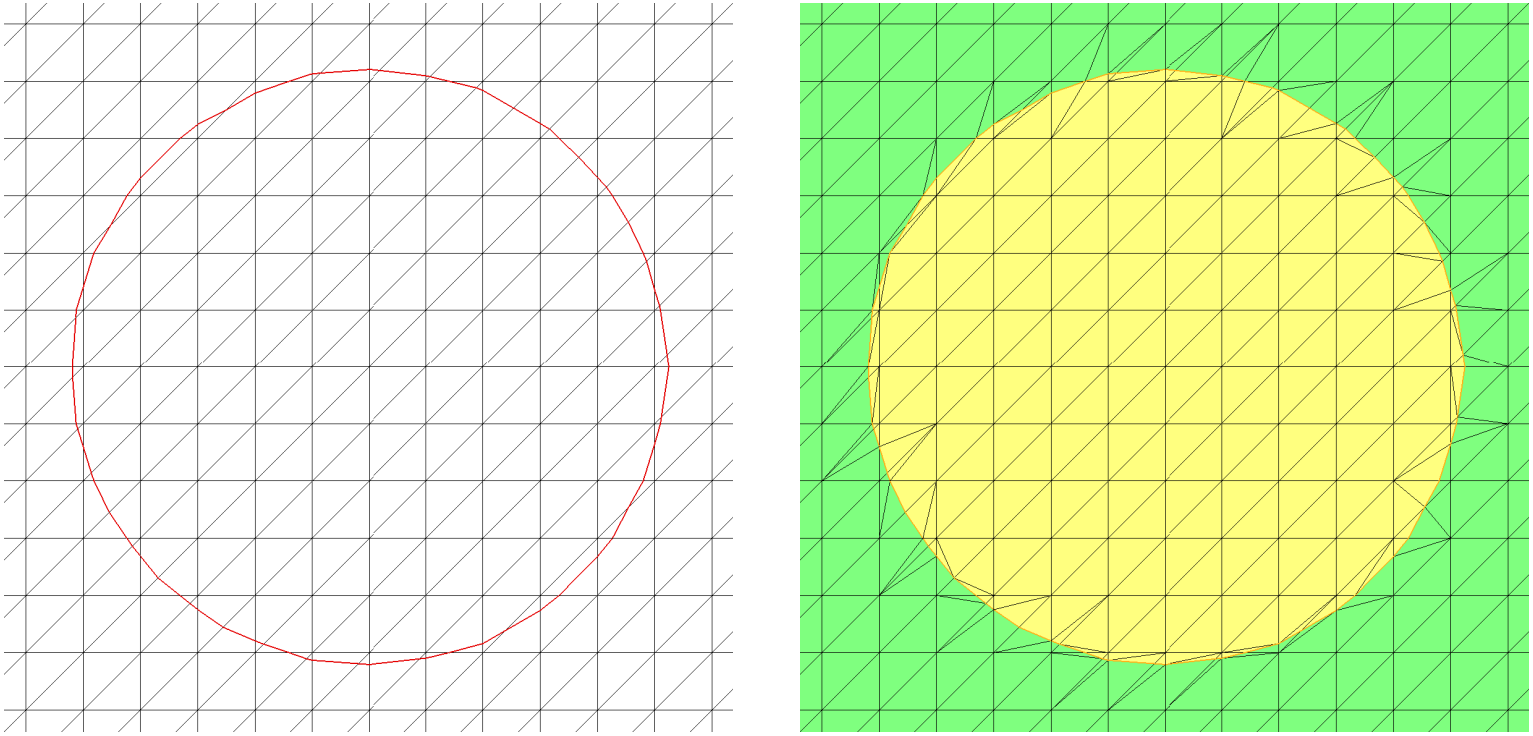


Figure 11: (left) 0 level set of a scalar function defined over a mesh ; (right) explicit discretization in the mesh.

However, doing so is bound to produce a very low-quality mesh, on which finite element computations will prove slow, inaccurate, not to say impossible.

Hence the need to improve the quality of the mesh while retaining its geometric features.

Local remeshing in $3d$

- Let $\mathcal{T}$ be an initial - valid, yet potentially ill-shaped - tetrahedral mesh $\mathcal{T}$. $\mathcal{T}$ carries a triangular surface mesh $\mathcal{S}_{\mathcal{T}}$, whose elements appear as faces of tetrahedra of $\mathcal{T}$.
- $\mathcal{T}$ is intended as an approximation of an ideal domain $\Omega \subset \mathbb{R}^3$, and $\mathcal{S}_{\mathcal{T}}$ as an approximation of its boundary $\partial\Omega$.

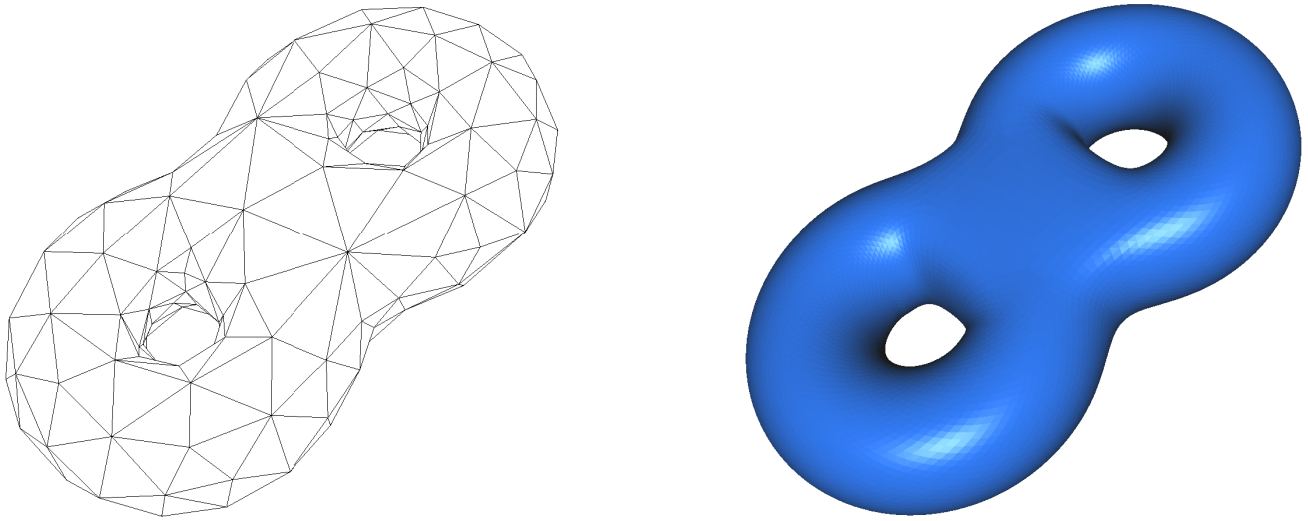


Figure 12: Poor geometric approximation (left) of a domain with smooth boundary (right)

Thanks to local mesh operations, we aim at getting a new, well-shaped mesh $\widetilde{\mathcal{T}}$, whose corresponding surface mesh $\mathcal{S}_{\widetilde{\mathcal{T}}}$ is a good approximation of $\partial\Omega$.

Local remeshing in $3d$: definition of an ideal domain

- In realistic cases, the ideal underlying domain Ω associated to $\mathcal{T}$ is unknown.
- However, from the sole data of $\mathcal{T}$ (and $\mathcal{S}_{\mathcal{T}}$), one can reconstruct approximations of geometric features of Ω: sharp angles, normal vectors at regular surface points,...
- These geometric data allow to define rules for the generation of a local parametrization of $\partial\Omega$, around a considered surface triangle $T \in \mathcal{S}_{\mathcal{T}}$, for instance as a Bézier surface.

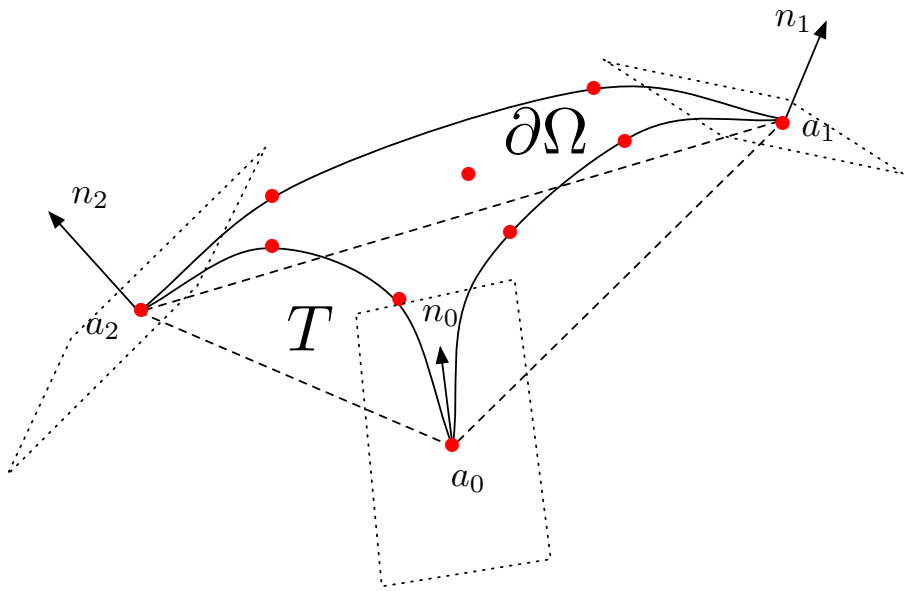


Figure 13: Generation of a cubic Bézier polynomial parametrization for the piece of $\partial\Omega$ associated to triangle T, from the approximated geometrical features (normal vectors at nodes).

Local mesh operators: edge splitting

If an edge pq is too long, insert its midpoint m, then split it into two.

- If pq belongs to a surface triangle $T \in \mathcal{S}_\mathcal{T}$, the midpoint m is inserted as the midpoint on the local piece of $\partial\Omega$ computed from T. Else, it is merely inserted as the midpoint of p and q.
- An edge may be 'too long' because it is too long when compared to the prescribed size, or because it causes a bad geometric approximation of $\partial\Omega$,...

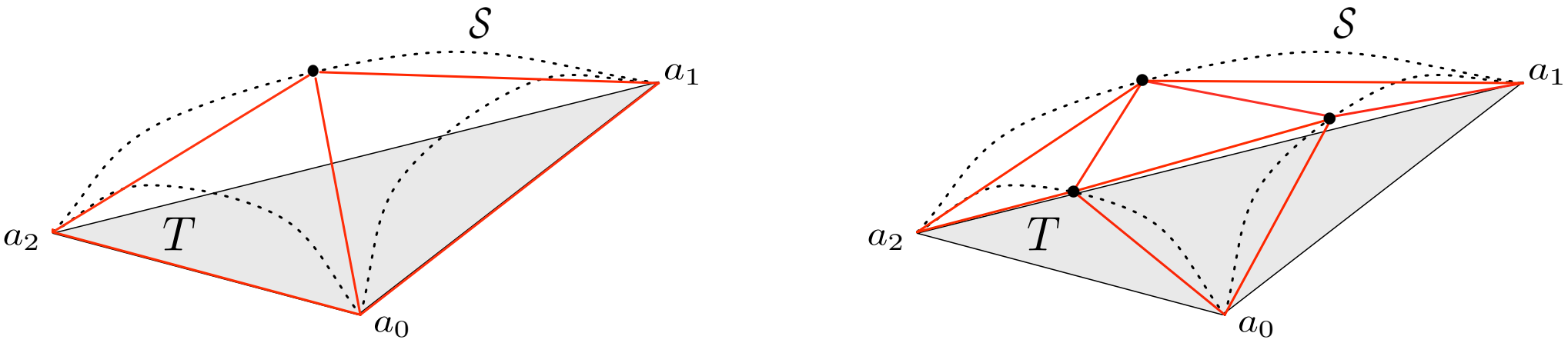


Figure 14: Splitting of one (left) or three (right) edges of triangle T, positioning the three new points on the ideal surface $\mathcal{S}$ (dotted).

Local mesh operators: edge collapse

If an edge pq is too short, merge its two endpoints.

- This operation may deteriorate the geometric approximation of $\partial\Omega$, and even invalidate some tetrahedra: some checks have to be performed to ensure the validity of the resulting configuration.
- An edge may be 'too short' because it is too long when compared to the prescribed size, or because it proves unnecessary to a nice geometric approximation of $\partial\Omega$,...

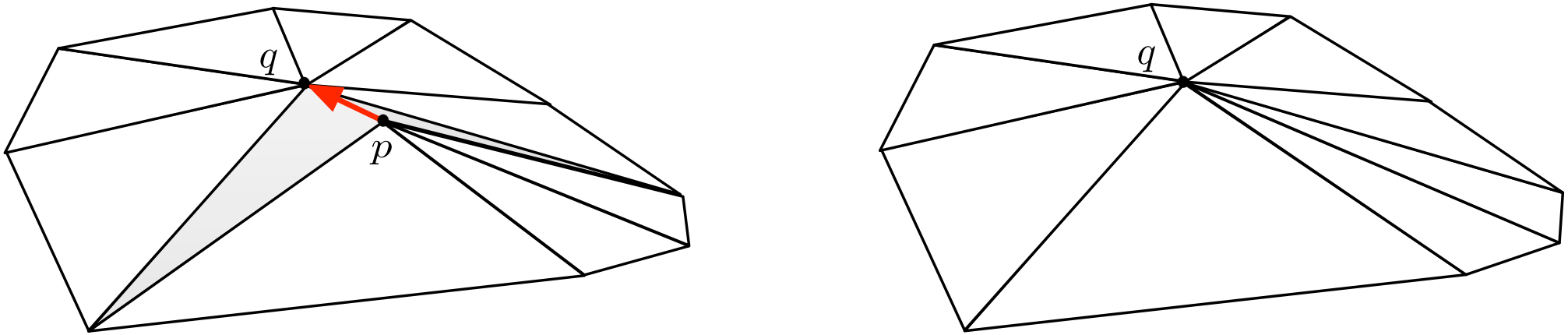


Figure 15: Collapse of point p over q.

Local mesh operators: edge swap

Some connectivities can be swapped in the mesh, for quality enhancement purposes.

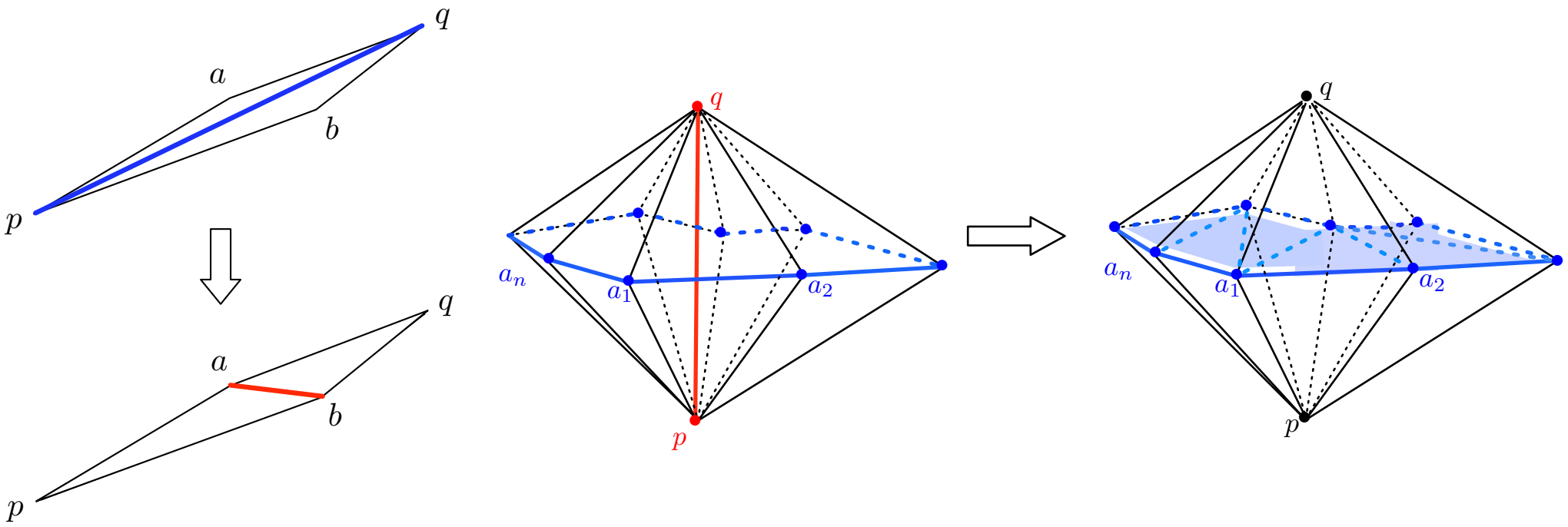


Figure 16: (left) $2d$ (or surface) swap of an edge pq; (right) $3d$ swap of an edge pq.

Local mesh operators: node relocation

Vertices can be relocated, in a one-by-one fashion, to increase the overall quality of the mesh.

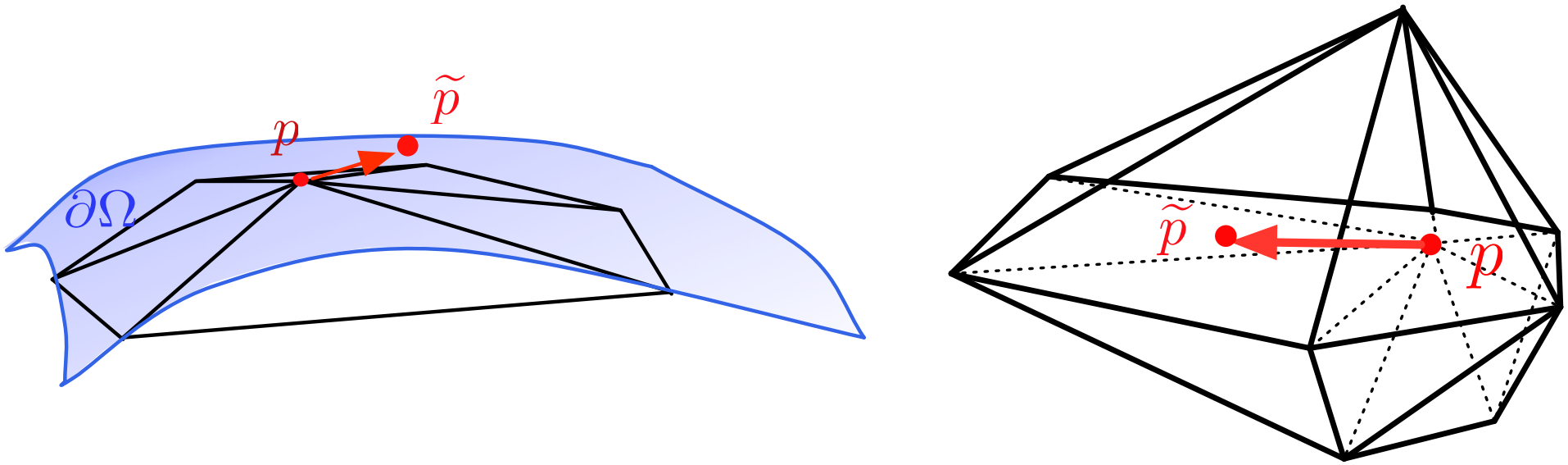


Figure 17: (left) Relocation of a surface vertex p; (right) relocation of an internal vertex p.

Local remeshing in $3d$: numerical examples

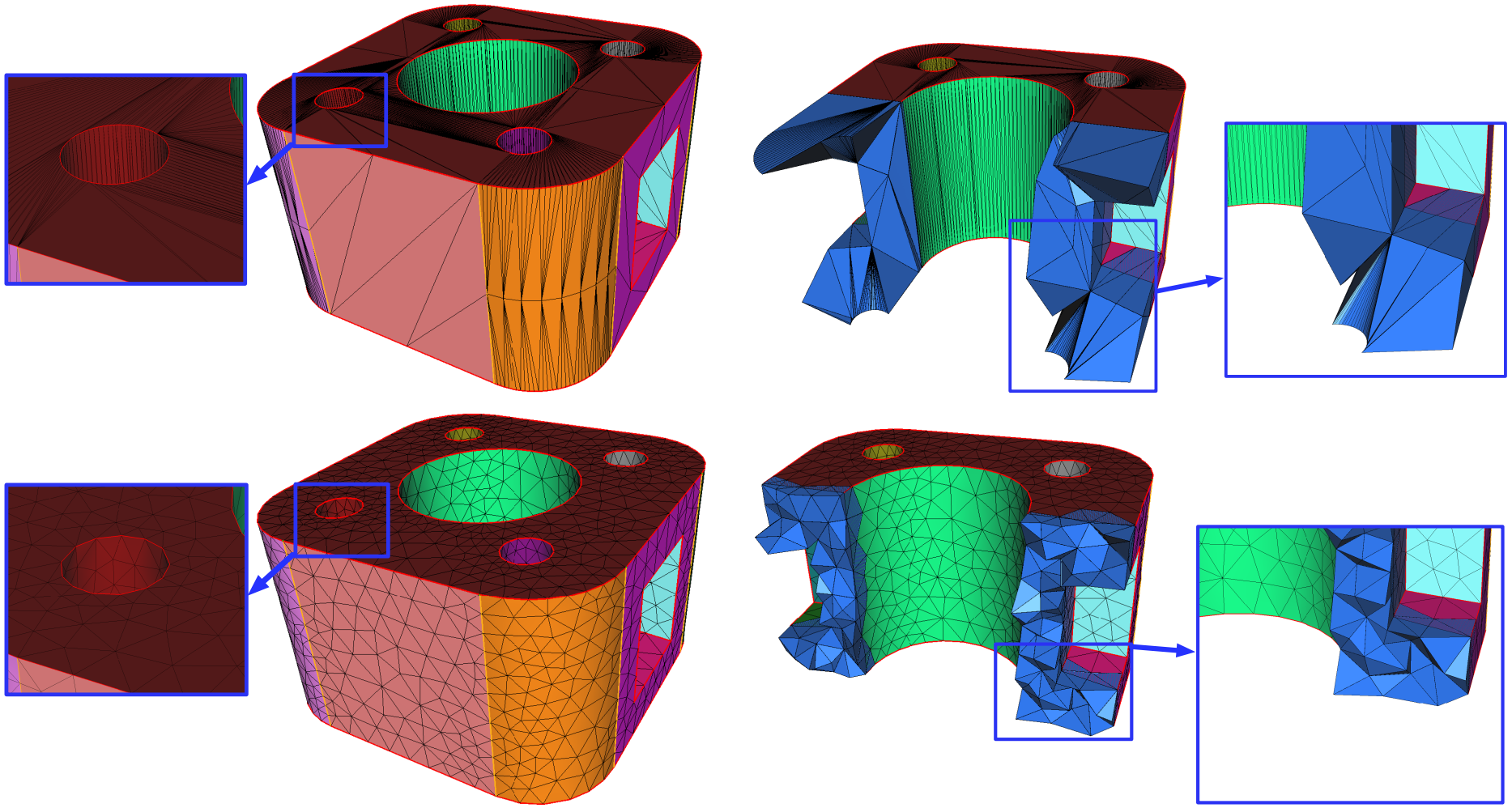


Figure 18: *Mechanical part before (left) and after (right) remeshing.*

Local remeshing in $3d$: numerical examples

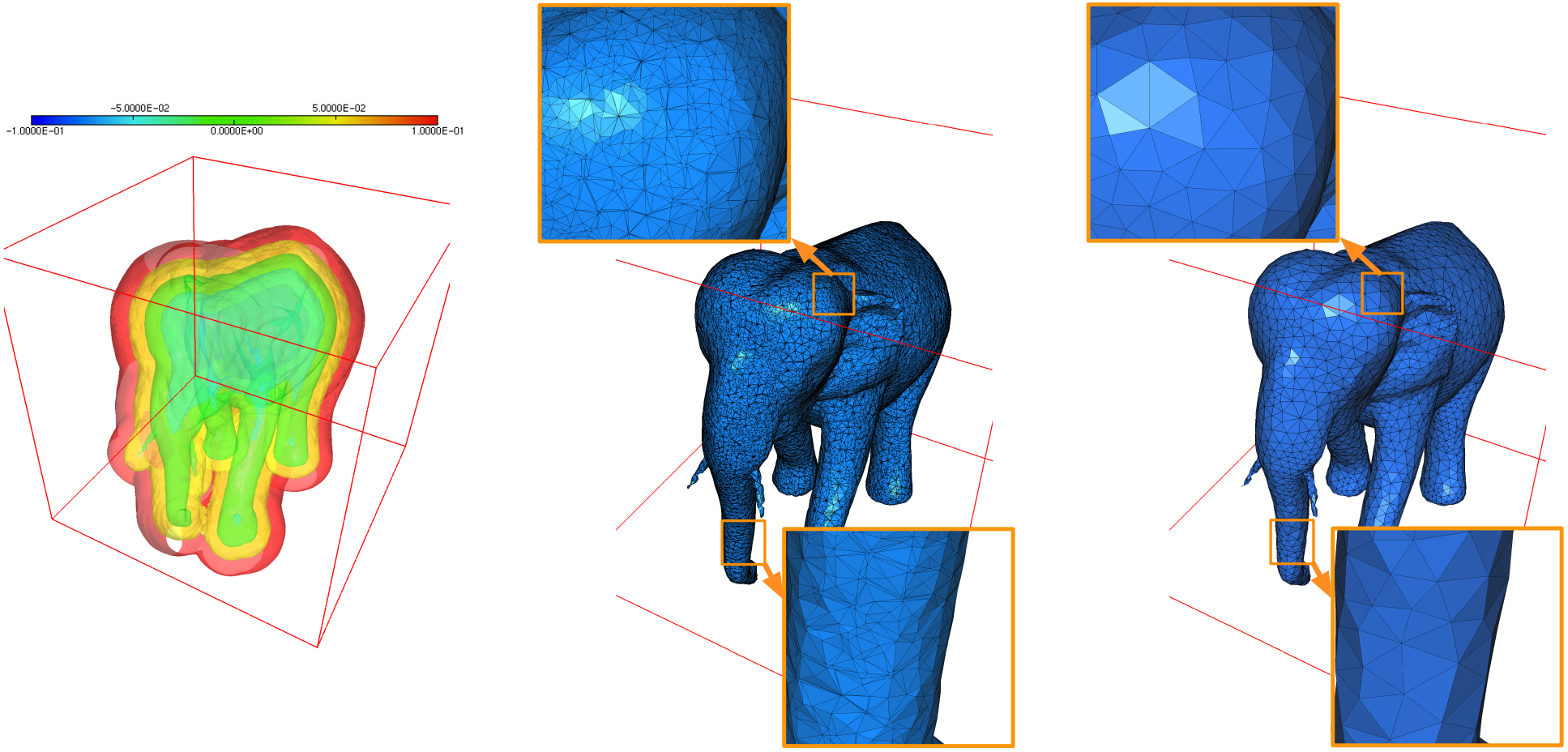


Figure 19: *(left) Some isosurfaces of an implicit function defined in a cube, (centre) result after rough discretization in the ambient mesh, (right) result after local remeshing.*

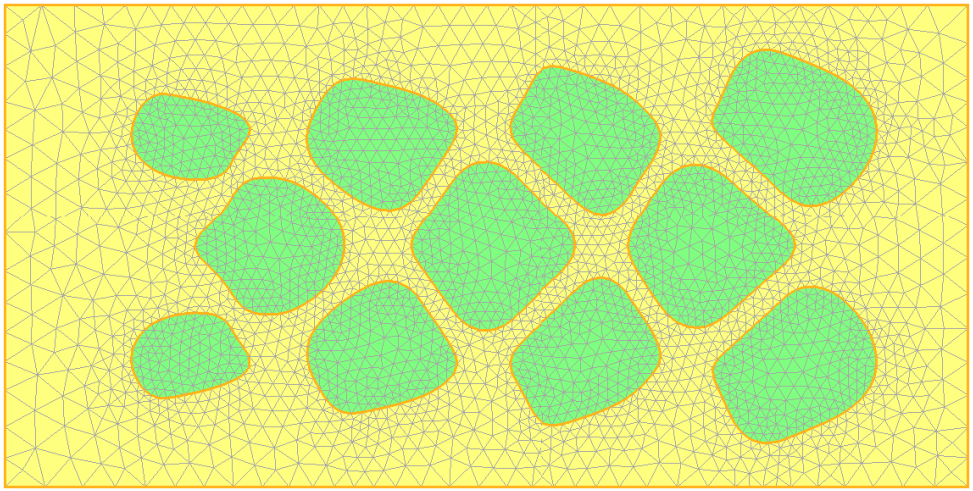
Part II.

A mesh evolution method for geometric shape optimization

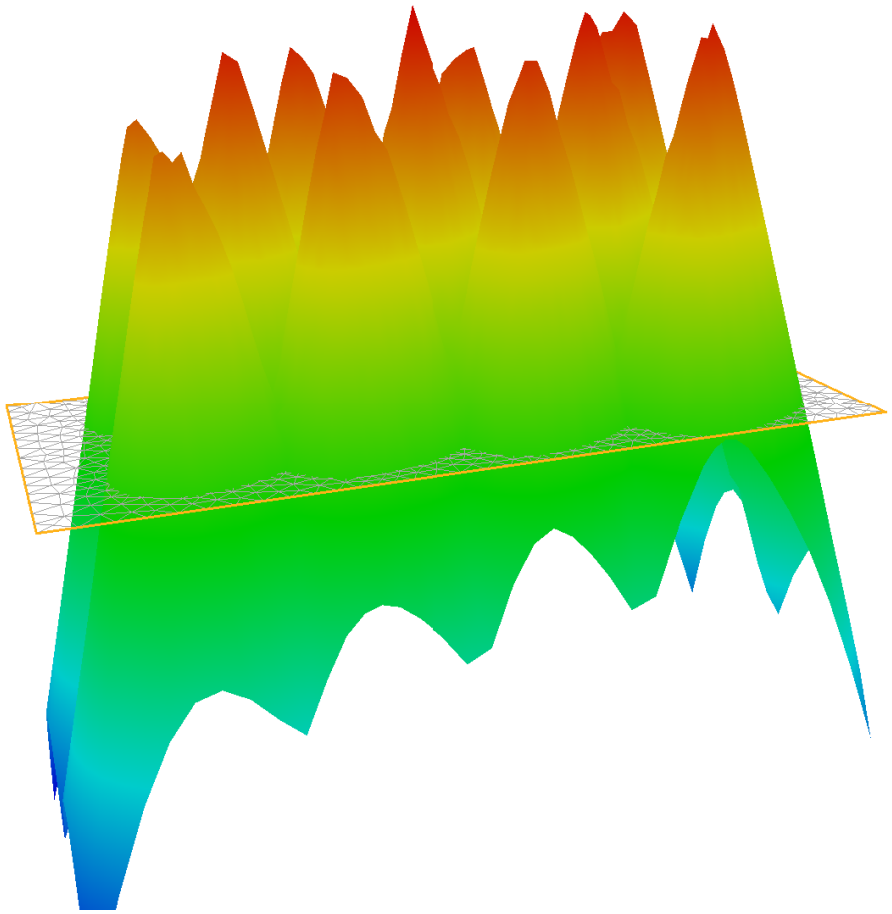
- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

The algorithm in action...

Step 1: Start with the actual shape Ω^n, and generate its signed distance function d_{Ω^n} over D, equipped with the mesh $\mathcal{T}^n$.



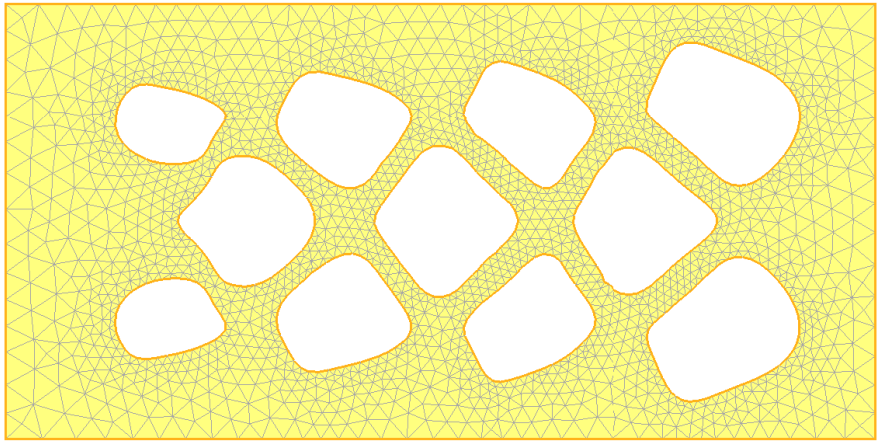
(a) The initial shape



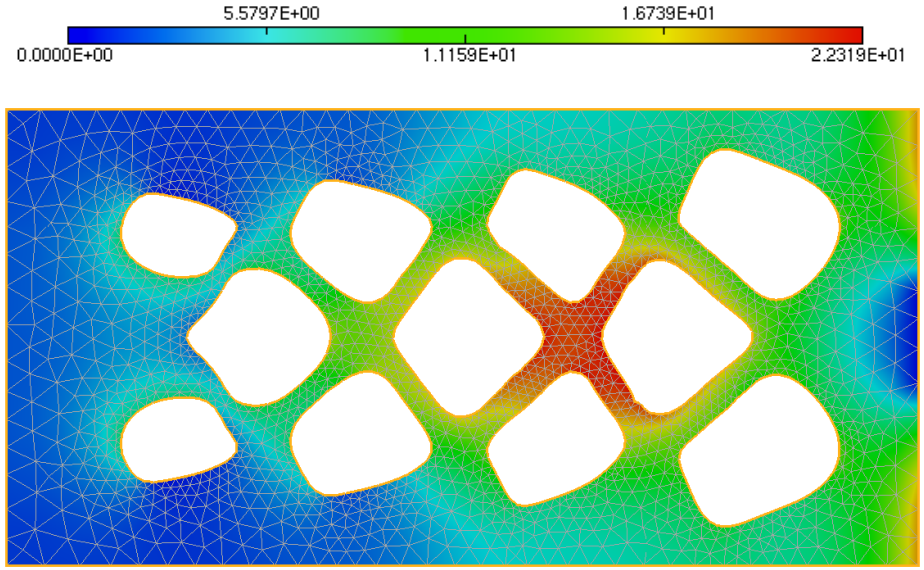
(b) Graph of d_{Ω^n}

The algorithm in action...

Step 2: "Forget" the exterior of the shape $D \setminus \Omega^n$, and perform the computation of the shape gradient $J'(\Omega^n)$ on (the mesh of) Ω^n.



(a) The "interior mesh"



(b) Computation of $J'(\Omega^n)$

The algorithm in action...

Step 3: "Remember" the whole mesh $\mathcal{T}^n$ of D. Extend the velocity field $J'(\Omega^n)$ to the whole mesh, and advect d_{Ω^n} along $J'(\Omega^n)$ for a (small) time step τ^n. A new level set function ϕ^{n+1} is obtained on $\mathcal{T}^n$, corresponding to the new shape Ω^{n+1}.

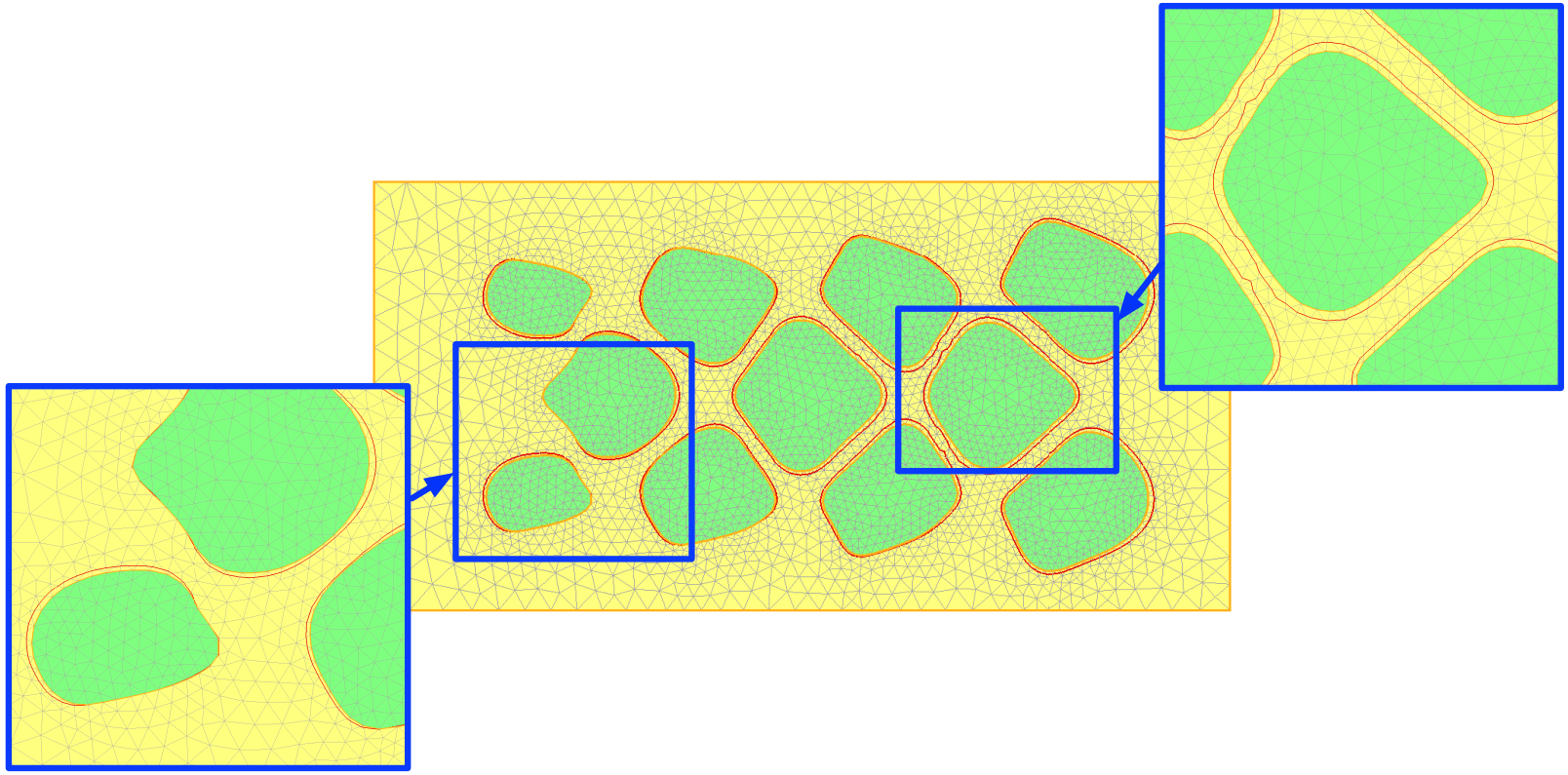


Figure 20: *The shape Ω^n, discretized in the mesh (in yellow), and the "new", advected 0-level set (in red).*

The algorithm in action...

Step 4: To close the loop, the 0 level set of ϕ^{n+1} is explicitly discretized in mesh $\mathcal{T}^n$. As expected, roughly "breaking" this line generally yields a very ill-shaped mesh.

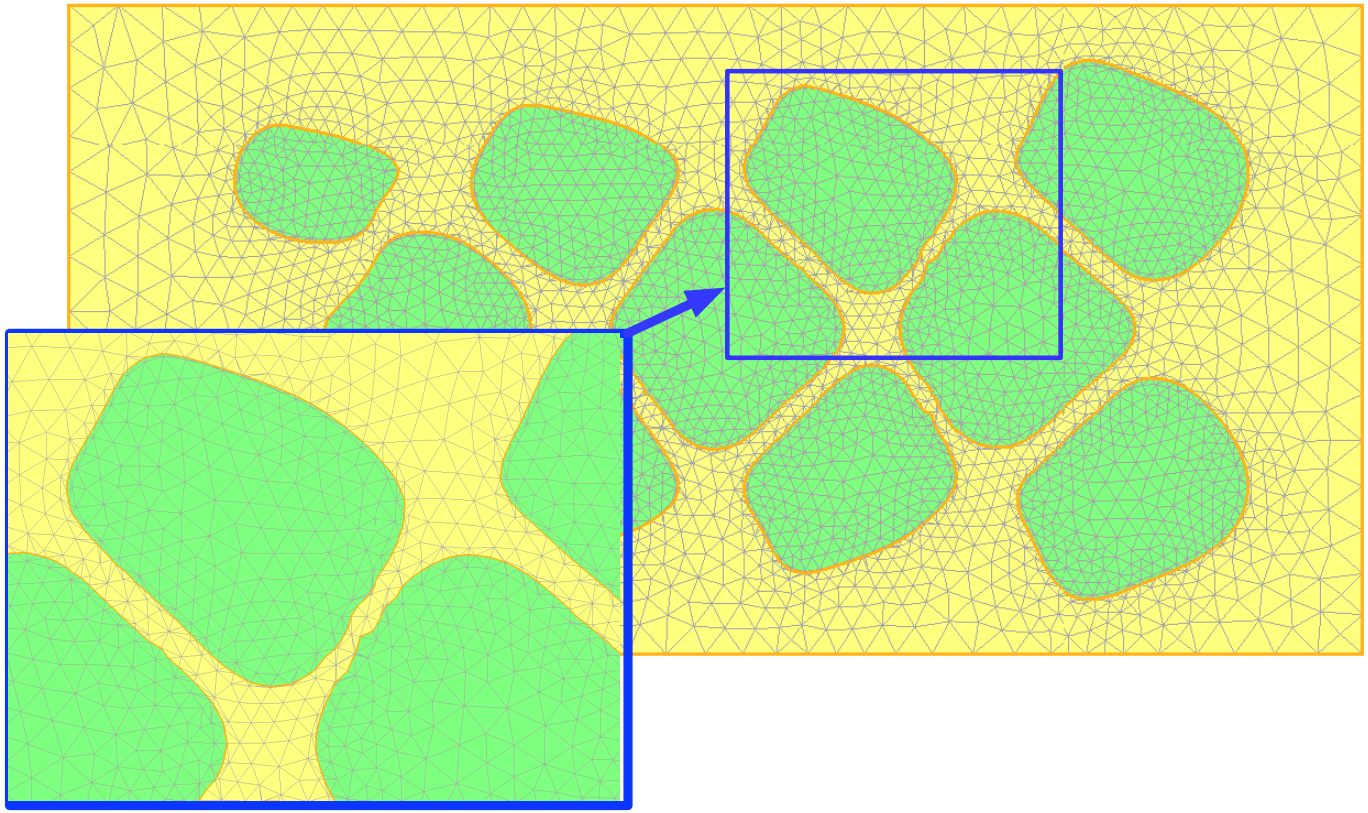


Figure 21: *Rough discretization of the* 0 *level set of* ϕ^{n+1} *into* $\mathcal{T}^n$*; the resulting mesh of* D *is ill-shaped.*

The algorithm in action...

The mesh modification step is then performed, so as to enhance the overall quality of the mesh according to the geometry of the shape. $\mathcal{T}^{n+1}$ is eventually obtained.

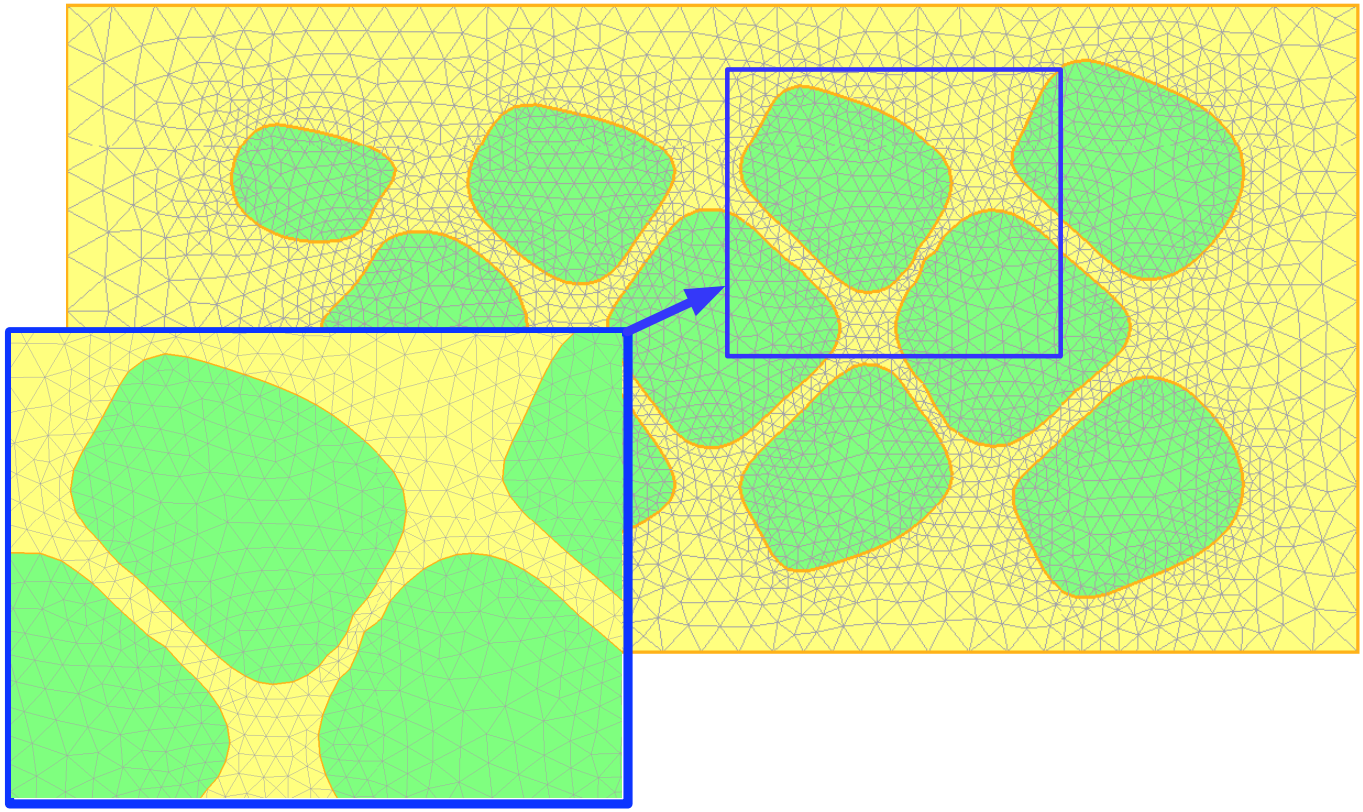
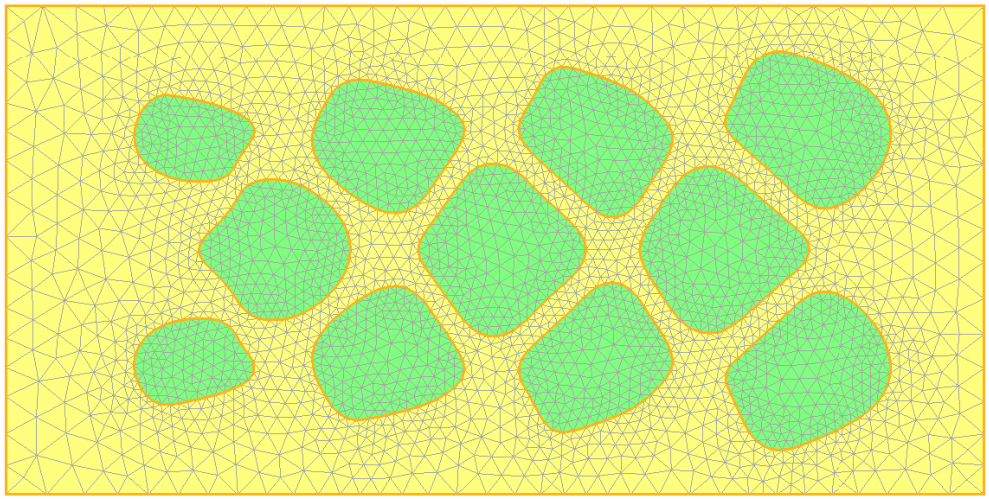
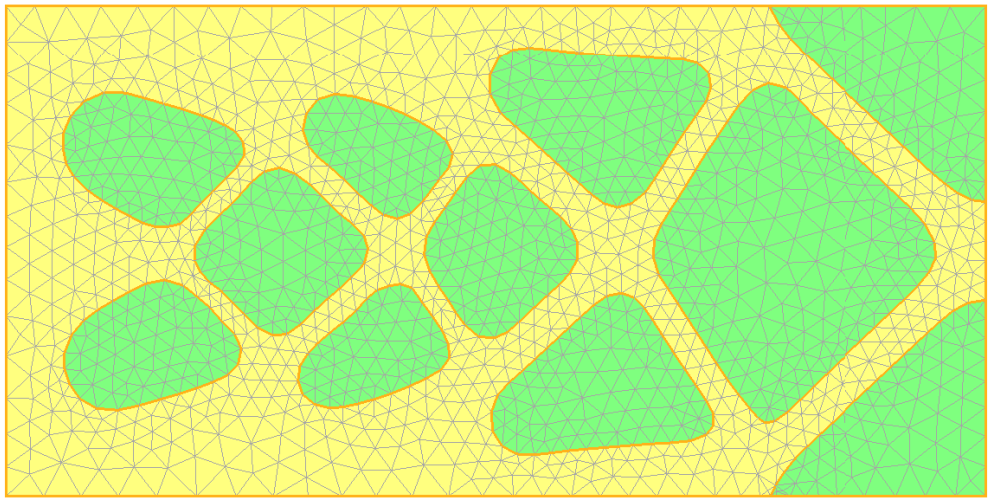


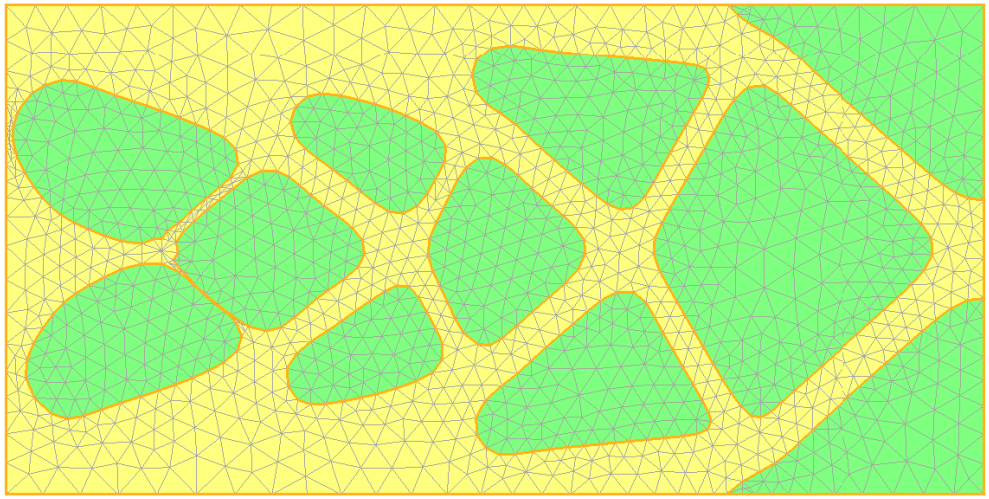
Figure 22: *Quality-oriented remeshing of the previous mesh ends with the new, well-shaped mesh $\mathcal{T}^{n+1}$ of D in which Ω^{n+1} is explicitly discretized.*

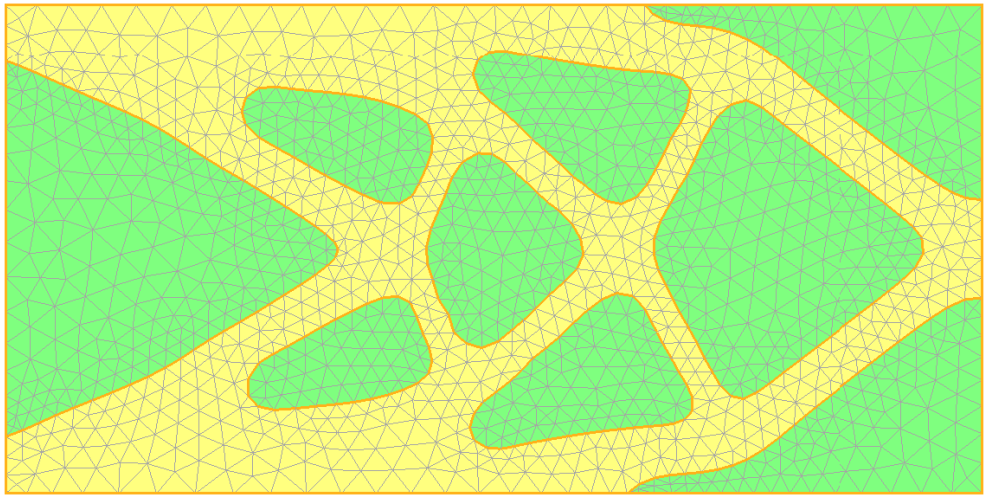
The algorithm in action...

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







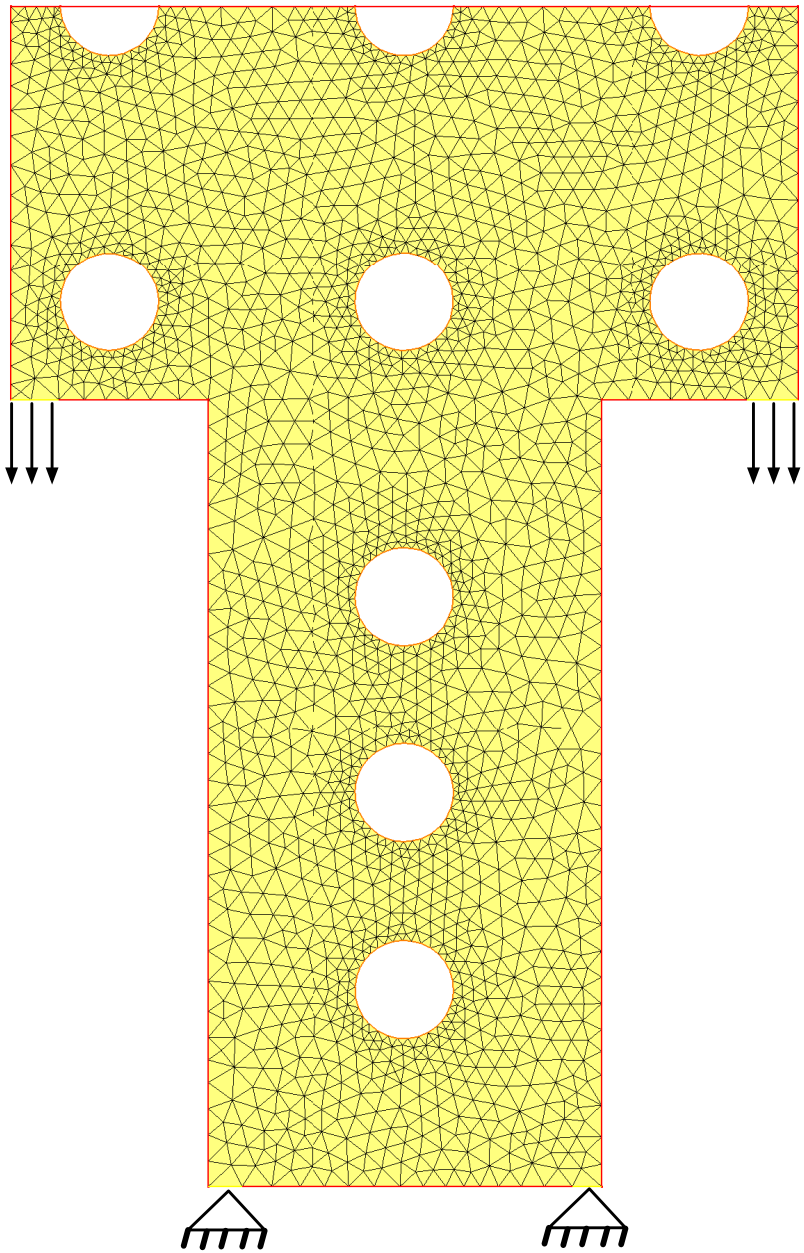


Part II.

A mesh evolution method for geometric shape optimization

- Presentation of the proposed method
- From a meshed to a level set description
- A meshing algorithm for implicit domains
- The method in action
- Numerical results

Numerical results: $2d$ optimal mast



The 'benchmark' two-dimensional optimal mast test case.

- Minimization of the compliance

$$C(\Omega) = \int_{\Omega} Ae(u_{\Omega}) : e(u_{\Omega})\, dx.$$

- A volume constraint is enforced by means of a fixed Lagrange multiplier.

Chaining fixed mesh and mesh evolution methods for shape optimization

- Starting from an initial shape Ω_0, the level set method for shape optimization is applied on a fixed mesh of D; a first 'optimal shape' $\widetilde{\Omega}$ is obtained.

- $\widetilde{\Omega}$ is explicitly discretized in the computational mesh, and serves as an initial shape for the mesh evolution method for shape optimization; a final 'optimal shape' Ω^* is obtained.

Numerical results: from one cantilever to another

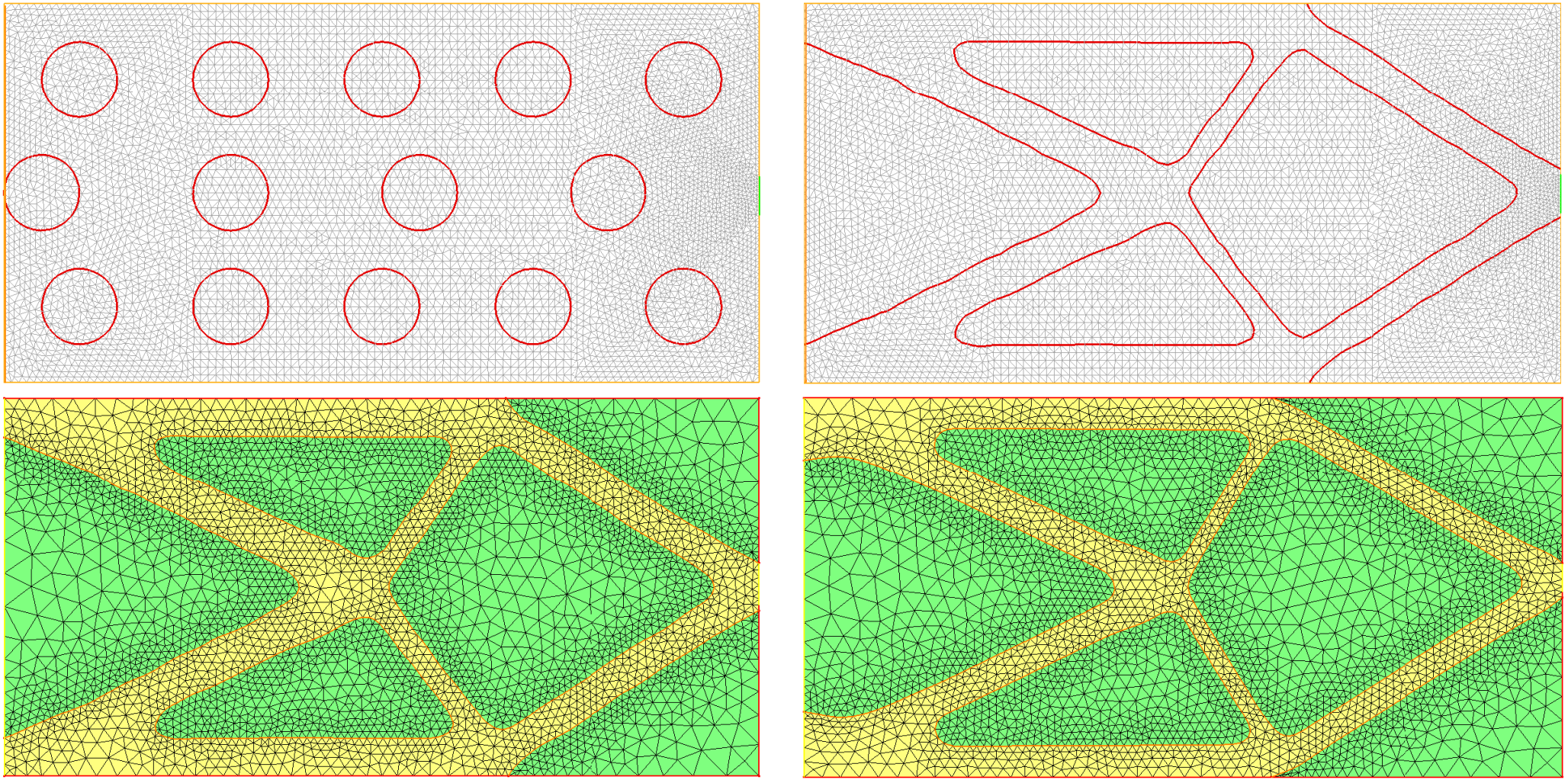


Figure 23: *(Top)* Initial and final iterations Ω_0 and $\widetilde{\Omega}$ of the $2d$ Cantilever test case, using the fixed mesh level set method. $C(\widetilde{\Omega}) + \ell \mathrm{Vol}(\widetilde{\Omega}) = 1.41$. *(Bottom-left)* Explicit discretization of $\widetilde{\Omega}$ into the computational mesh: $C(\widetilde{\Omega}) + \ell \mathrm{Vol}(\widetilde{\Omega}) = 1.63$; *(Bottom-right)* final shape Ω^*. $C(\Omega^*) + \ell \mathrm{Vol}(\Omega^*) = 1.09$.

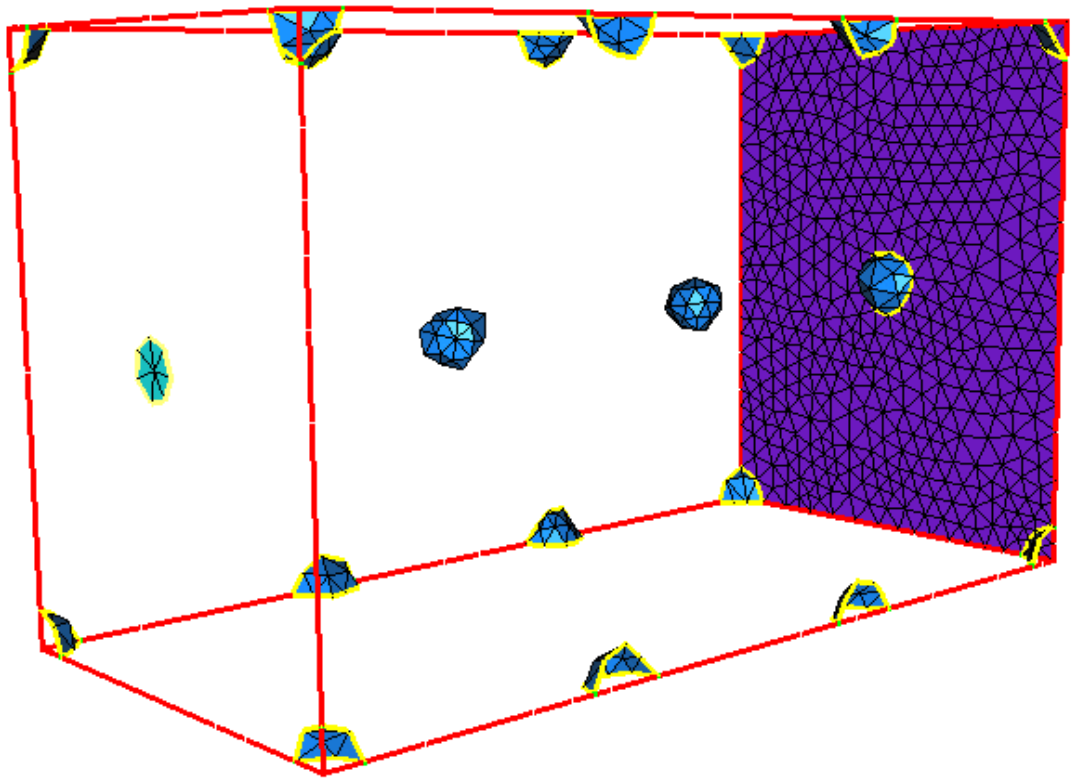
Numerical results: $3d$ cantilever

The 'benchmark' three-dimensional cantilever test case.

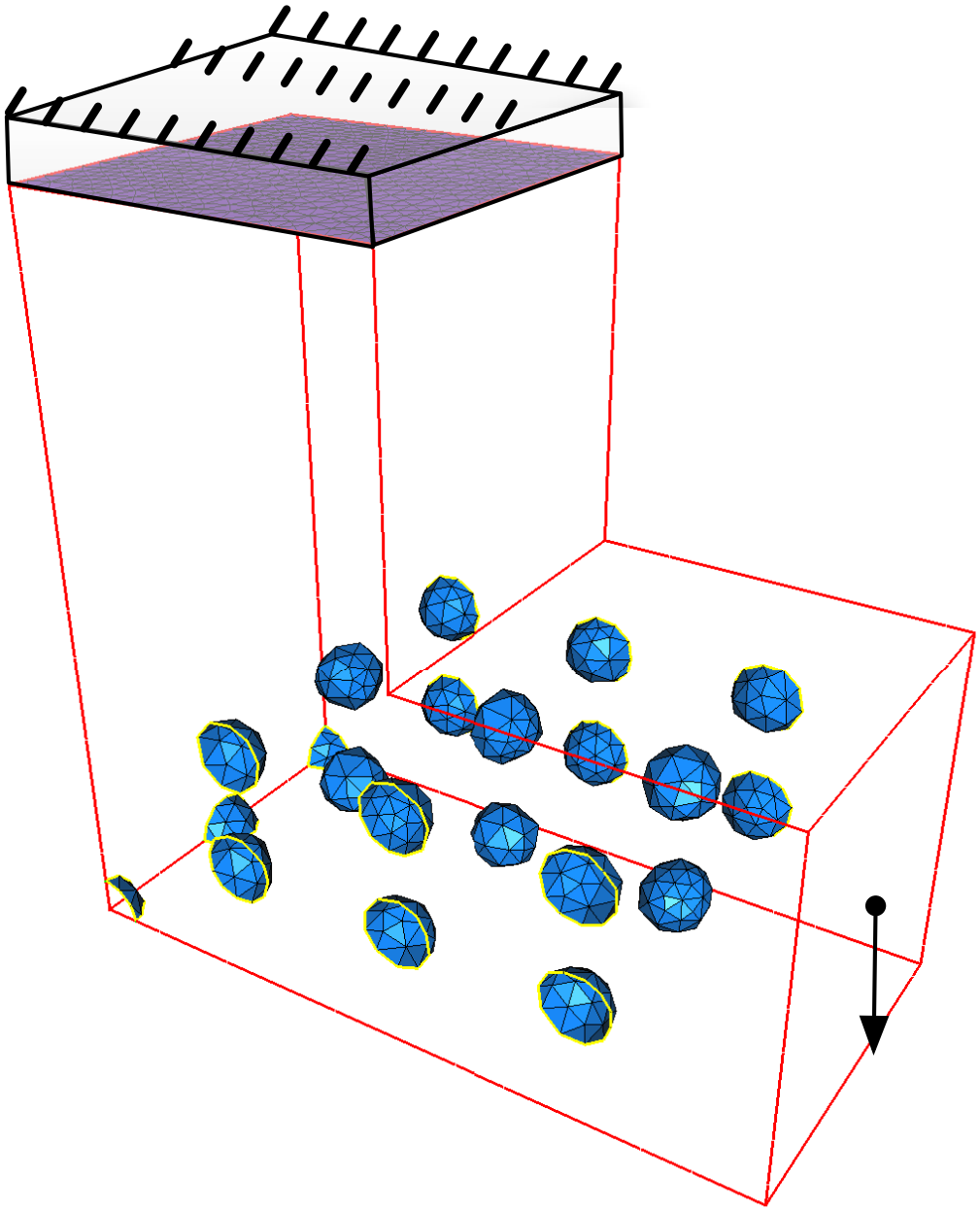
- Minimization of the compliance

$$C(\Omega) = \int_\Omega Ae(u_\Omega) : e(u_\Omega)\, dx.$$

- A volume constraint is enforced by means of a fixed Lagrange multiplier.



Numerical results: 3d L-Beam



Optimal design of a L-shaped beam.

• Minimization of a stress-based criterion

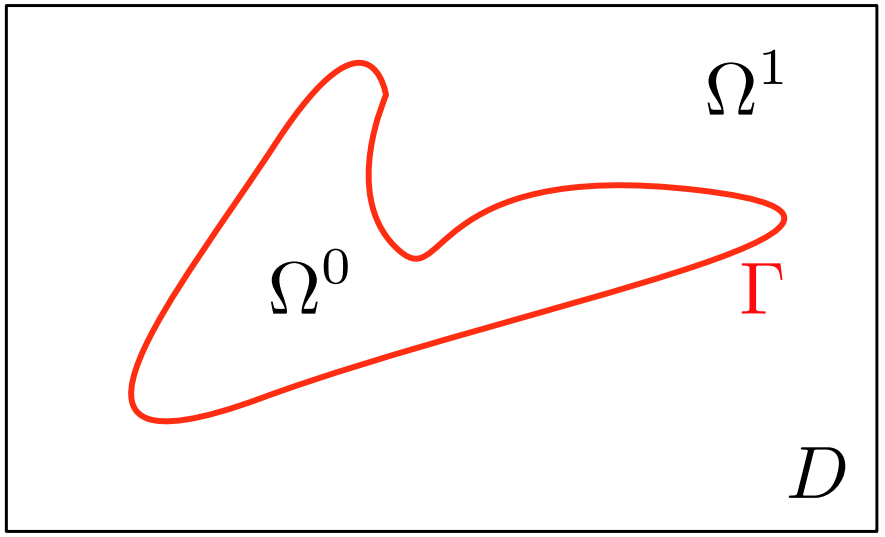
$$S(\Omega) = \int_\Omega k(x)||\sigma(u_\Omega)||^2 \, dx,$$

where k is a weight factor, and $\sigma(u) = Ae(u)$ is the stress tensor.

• A volume constraint is enforced by means of a fixed Lagrange multiplier.

Another example in multi-phase optimization

Optimal repartition of two materials A_0, A_1 occupying subdomains Ω^0 and $\Omega^1 := D \setminus \Omega^0$ of a fixed working domain D, with total (discontinuous) Hooke's law $A_{\Omega^0} := A_0\chi_{\Omega^0} + A_1\chi_{\Omega^1}$.



- Minimization of the total compliance of D: $C(\Omega^0) = \int_D A_{\Omega^0} e(u_{\Omega^0}) : e(u_{\Omega^0})\, dx$.
- Shape derivative (see [Allaire, Jouve, Van Goethem]):

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

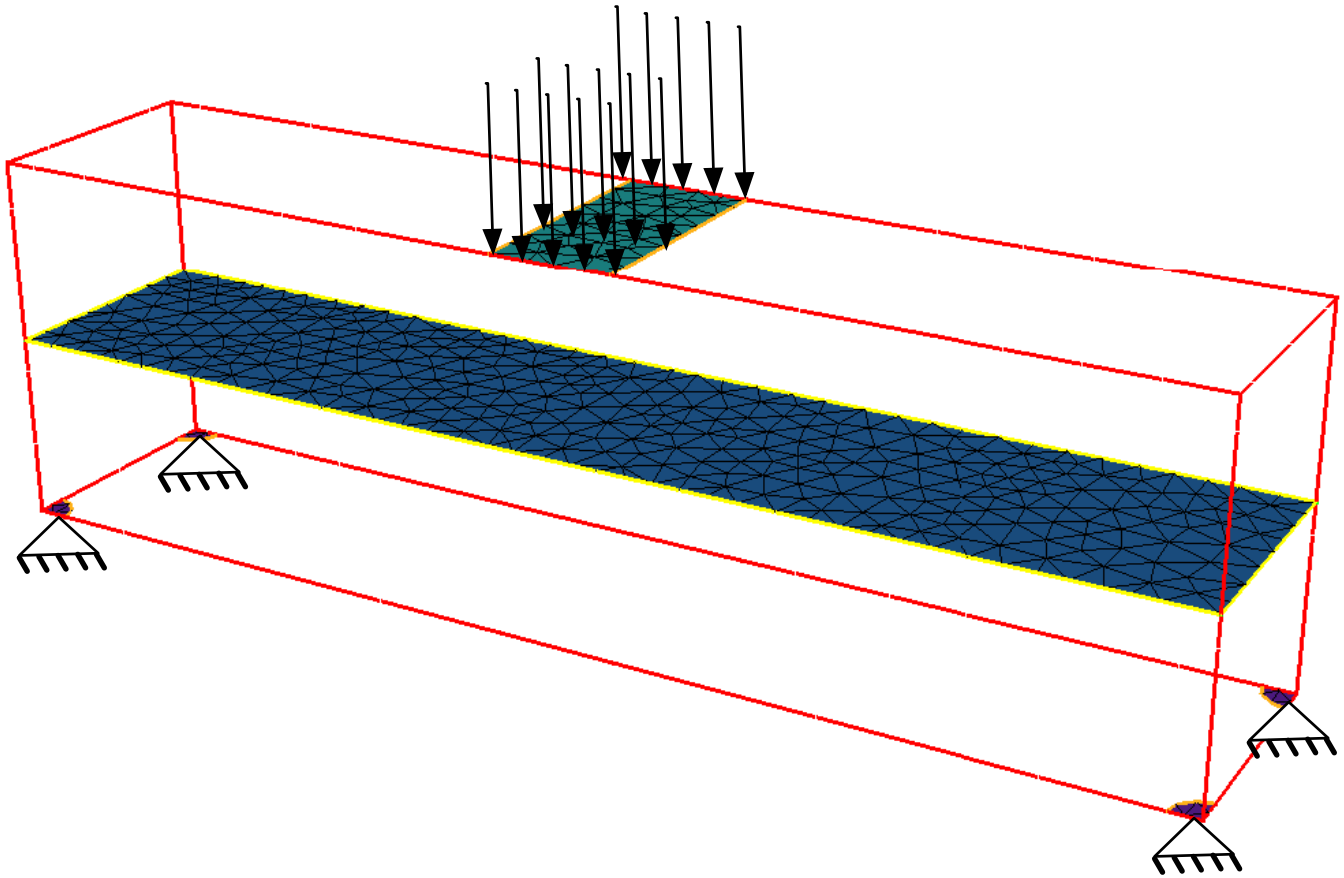
- Evaluating $\mathcal{D}(u,u)$ is a bit awkward in a fixed mesh context, for it involves jumps of the (discontinuous) strain and stress tensors $e(u)$ and $\sigma(u)$ at the interface Γ.

  In this setting, one has to resort to approximations of the problem (e.g. by differentiating the discrete finite element problem, or by using a smoothed-interface approximation).

Numerical results: a multi-phase beam

Minimization of the compliance of a beam D, consisting of a mixture of a material A_0 with another one A_1, with Young's modulus $E^1 = E^0/3$.

A constraint on the volume of the stronger material is enforced by means of a fixed Lagrange multiplier.



Perspectives

Perspectives

Part I.

- The method presented in Part I. is a general method for approximating a supremal functional. It could also be used to deal with fiability constraints, i.e. constraints of the form:

$$\max_{\delta \in \mathcal{P}} c(u(h, \delta)) \leq \bar{c}.$$

Part II.

- Robustification of the proposed numerical method for tackling the issue of mesh deformation in the shape optimization context, which is in the course of being integrated in an industrial environment (RODIN project).
- Some new ingredients could be added to the general strategy, e.g. a way of denoising the successive shapes Ω^n.
- In the final steps of the process, when only very 'small' changes are expected from one iteration to the next, it could be interesting to switch to a moving node strategy. On the theoretical side, is there a way to cook the velocity field θ^n so that moving the mesh of Ω^n is always valid ?
- Application of the numerical method for mesh evolution to other surface evolution problems (e.g. in fluid dynamics, fluid-structure interactions, solidification problems, etc...).

Thank you !