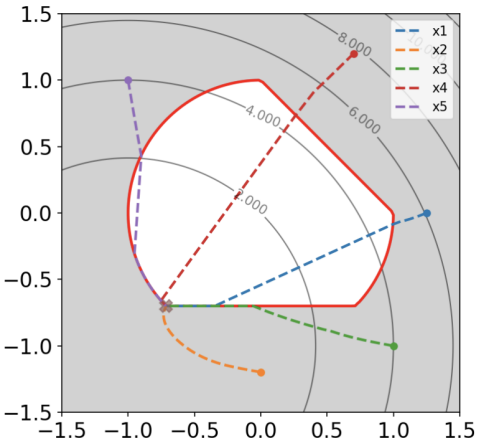
The level set method for structural topology optimization: Part II

Grégoire Allaire[1], Charles Dapogny[2]

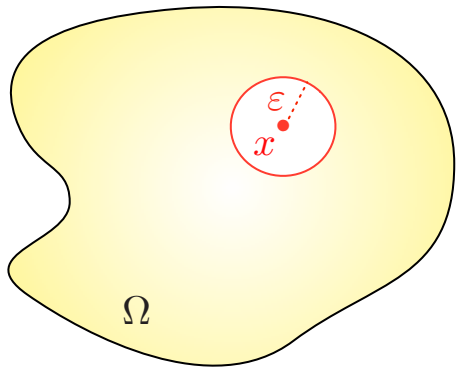
[1] Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France
[2] Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France

20th May, 2026

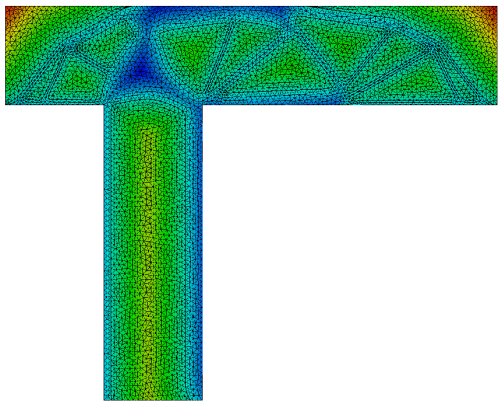
- The calculation of shape derivatives.
- Advanced practice of the Level Set Method.



The null-space algorithm.

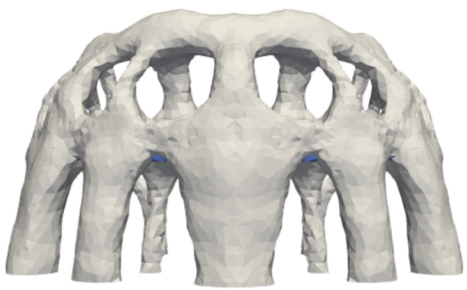


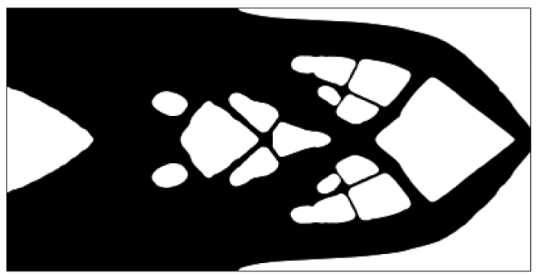
Topological derivatives.

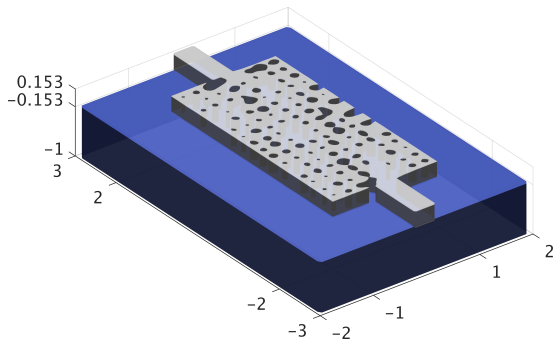


The Level Set Method on a triangular mesh.

- Further numerical experiments.







Part I

Shape derivatives

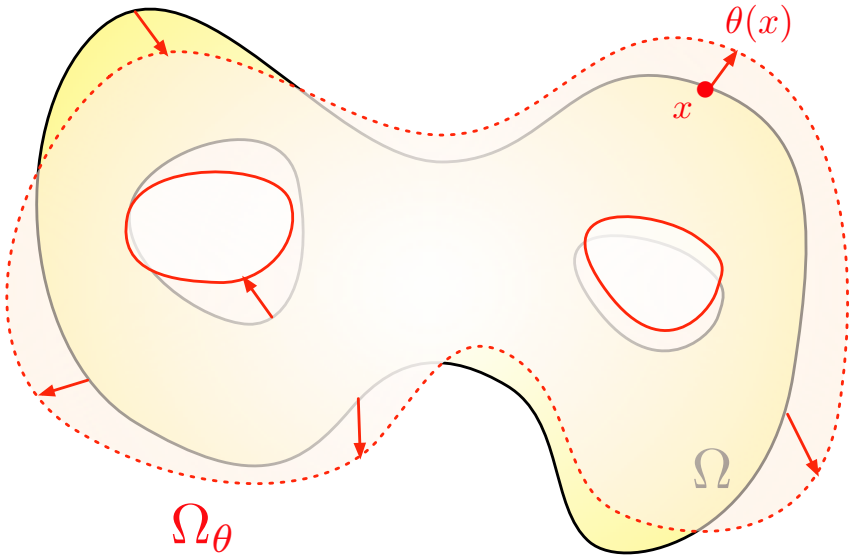
Reminder of Webinar 1 (I)

In Webinar 1, we have introduced:

- Hadamard's method for describing variations of a shape Ω of the form:

$$\Omega_\theta := (\mathrm{Id} + \theta)(\Omega),$$

for "small" vector fields $\theta : \mathbb{R}^d \to \mathbb{R}^d$.



- The shape derivative $J'(\Omega)(\theta)$ of a function of the domain $J(\Omega)$, through the Taylor expansion:

$$J(\Omega_\theta) = J(\Omega) + J'(\Omega)(\theta) + \mathrm{o}\left(||\theta||\right).$$

Reminder of Webinar 1 (II)

- We have discussed the shape derivatives of "simple" objective functions:
  - If $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth function,

$$J(\Omega) := \int_\Omega f(x)\mathrm{d}x \text{ has shape derivative } J'(\Omega)(\theta) = \int_{\partial\Omega} f\, \theta \cdot n \,\mathrm{d}s;$$

  - If $g : \mathbb{R}^d \to \mathbb{R}$ is a smooth function,

$$J(\Omega) := \int_{\partial\Omega} g(x)\mathrm{d}s(x), \text{ then } J'(\Omega)(\theta) = \int_{\partial\Omega} \left(\frac{\partial g}{\partial n} + \kappa g\right) \theta \cdot n \,\mathrm{d}s.$$

- We now aim to calculate shape derivatives of "physical" functionals, depending on Ω via the solution of PDE posed on Ω.

- Céa's method used to this end is a shape optimization instance of the celebrated adjoint method.

A review of the adjoint method (I)

Let us consider the following optimization problem:

$$\min_{\rho \in \mathcal{U}_{\mathrm{ad}}} J(\rho) = j(\rho, u_\rho),$$

where

- The set $\mathcal{U}_{\mathrm{ad}}$ of admissible designs ρ is a subset of $\mathbb{R}^k$ for some $k \geq 1$;
- For given $\rho \in \mathcal{U}_{\mathrm{ad}}$, the state $u_\rho \in \mathbb{R}^n$ is the solution to the system

$$A(\rho) u_\rho = b,$$

  where $b \in \mathbb{R}^n$ and $A(\rho)$ is an invertible $n \times n$ matrix.

The adjoint method is a systematic way to calculate the derivative of $J(\rho)$.

References: See

- [Li] for the seminal idea,
- [Err, GiPier, Ple] for comprehensive, application-oriented introductions.

A review of the adjoint method (I)

- This task relies on the Lagrangian functional $\mathcal{L}(\rho, u, p)$:

$$\text{For } \rho \in \mathcal{U}_{\text{ad}},\ u \in \mathbb{R}^n \text{ and } p \in \mathbb{R}^n, \quad \mathcal{L}(\rho, u, p) = j(\rho, u) + p \cdot (A(\rho)u - b)\,.$$

- For each $\rho \in \mathcal{U}_{\text{ad}}$, the state u_ρ is the solution to:

$$\frac{\partial \mathcal{L}}{\partial p}(\rho, u_\rho, p_\rho) = 0.$$

- We introduce the adjoint state p_ρ as the solution to:

$$\frac{\partial \mathcal{L}}{\partial u}(\rho, u_\rho, p_\rho) = 0, \text{ that is: } A(\rho)^T p_\rho = -\frac{\partial j}{\partial u}(\rho, u_\rho).$$

A review of the adjoint method (II)

Theorem.

The derivative of $J(\rho)$ reads:

$$J'(\rho) = \frac{\partial \mathcal{L}}{\partial \rho}(\rho, u_\rho, p_\rho) = \frac{\partial j}{\partial \rho}(\rho, u_\rho) + p_\rho \cdot A'(\rho) u_\rho.$$

Proof:

- Since the state u_ρ satisfies the constraint $A(\rho)u_\rho = b$, we have:

$$\text{For any } \rho \in \mathcal{U}_{\text{ad}}, \quad J(\rho) = \mathcal{L}(\rho, u_\rho, p).$$

- Taking derivatives with respect to ρ and using the chain rule yields:

$$J'(\rho)(\widehat{\rho}) = \frac{\partial \mathcal{L}}{\partial \rho}(\rho, u_\rho, p)(\widehat{\rho}) + \frac{\partial \mathcal{L}}{\partial u}(\rho, u_\rho, p)\left(\frac{\partial u_\rho}{\partial \rho}(\widehat{\rho})\right),$$

  featuring the "difficult" derivative $\frac{\partial u_\rho}{\partial \rho}(\widehat{\rho})$ of the mapping $\rho \mapsto u_\rho$.

- By definition of the adjoint state, setting $p = p_\rho$ in this identity cancels the last term and proves the result.

Céa's method for shape optimization (I)

Let us now turn to the geometric optimization problem

$$\min_{\Omega \in \mathcal{U}_{\mathrm{ad}}} J(\Omega) = \int_{\Omega} j(u_\Omega)\,\mathrm{d}x \qquad (\mathcal{P})$$

where

- $\mathcal{U}_{\mathrm{ad}}$ is a set of admissible shapes Ω;
- For given $\Omega \in \mathcal{U}_{\mathrm{ad}}$, the state u_Ω is the solution to an equation:

$$\mathtt{PDE}(u_\Omega) = 0 \quad \text{in } \Omega.$$

In the spirit of the adjoint method, Céa's method relies on the Lagrangian:

For all $\Omega \in \mathcal{U}_{\mathrm{ad}}$ and functions u, p,

$$\mathcal{L}(\Omega, u, p) = \underbrace{\int_{\Omega} j(u)\,\mathrm{d}x}_{\text{Objective function at stake}} + \underbrace{\mathrm{VF}(\Omega, u, p).}_{\substack{\text{Variational Formulation of the PDE} \\ \text{with a test function or Lagrange multiplier } p}}$$

Céa's method for shape optimization (II)

- The shape Ω and the variables (u, p) featured in the definition of $\mathcal{L}(\Omega, u, p)$ must be independent.

- Again, the state u_Ω and the adjoint p_Ω make up a saddle point of the Lagrangian:

$$u_\Omega \text{ is characterized by: } \forall \widehat{p}, \quad \frac{\partial \mathcal{L}}{\partial p}(\Omega, u_\Omega, p_\Omega)(\widehat{p}) = 0,$$

and

$$p_\Omega \text{ is characterized by: } \forall \widehat{u}, \quad \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega)(\widehat{u}) = 0.$$

.

- The derivative of the objective function $J(\Omega)$ follows as:

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

- The method is formal, as it assumes that the mapping $\Omega \mapsto u_\Omega$ is differentiable.

Céa's method for shape optimization (III)

We exemplify the use of Céa's method in shape optimization with 3 situations:

1. The PDE at stake is the Laplace equation with Neumann boundary condition; the complete calculation is described.
2. The PDE is the Laplace equation with Dirichlet boundary condition.
3. We consider an elliptic equation featuring two material phases, whose interface is optimized, instead of the exterior boundary.

G. Allaire, C. Dapogny, and F. Jouve, *Shape and topology optimization*, in Geometric partial differential equations, part II, A. Bonito and R. Nochetto eds., Handbook of Numerical Analysis, vol. 22, (2021), pp.1-132.

Céa's method: the Neumann case (I)

- To set ideas, we consider the Laplace equation with Neumann boundary conditions:

$$\begin{cases} -\Delta u + u = f & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases}$$

Here, the $+u$ term is added to ensure that the system is well-posed in $H^1(\Omega)$.

- The Lagrangian functional associated to the problem $(\mathcal{P})$ reads:

For any shape $\Omega \in \mathcal{U}_{\mathrm{ad}}$, $u, p \in H^1(\mathbb{R}^d)$, $\quad \mathcal{L}(\Omega, u, p) =$

$$\underbrace{\int_\Omega j(u)\,\mathrm{d}x}_{\substack{\textbf{Objective function} \\ \textbf{where } u_\Omega \textbf{ is replaced by } u}} + \underbrace{\int_\Omega \nabla u \cdot \nabla p\,\mathrm{d}x + \int_\Omega up\,\mathrm{d}x - \int_\Omega fp\,\mathrm{d}x}_{\substack{\textbf{Penalization of the "constraint" } u=u_\Omega: \\ \int_\Omega (-\Delta u+u-f)p\,\mathrm{d}x=0}}.$$

- The variables $\Omega \in \mathcal{U}_{\mathrm{ad}}$ and $u, p \in H^1(\mathbb{R}^d)$ are independent.

Céa's method: the Neumann case (II)

We look for the saddle points of the Lagrangian $(u, p) \mapsto \mathcal{L}(\Omega, u, p)$:

- By construction, for all $\widehat{p} \in H^1(\mathbb{R}^d)$,

$$\frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p)(\widehat{p}) = \int_\Omega \nabla u \cdot \nabla \widehat{p} \, \mathrm{d}x + \int_\Omega u \widehat{p} \, \mathrm{d}x - \int_\Omega f \widehat{p} \, \mathrm{d}x.$$

  The variational formulation of the state equation is given by:

$$\frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p)(\widehat{p}) = 0.$$

  This implies that $u \equiv u_\Omega$.

- The adjoint state p_Ω is characterized by:

$$\forall \widehat{u} \in H^1(\mathbb{R}^d), \quad \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega)(\widehat{u}) = 0.$$

  An elementary computation yields

$$\frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p)(\widehat{u}) = \int_\Omega j'(u)\widehat{u} \, \mathrm{d}x + \int_\Omega \nabla \widehat{u} \cdot \nabla p \, \mathrm{d}x + \int_\Omega \widehat{u} p \, \mathrm{d}x,$$

  and we deduce that p_Ω is solution to:

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

Céa's method: the Neumann case (III)

Theorem.

The shape derivative of the objective function $J(\Omega)$ is:

$$\forall \theta : \mathbb{R}^d \to \mathbb{R}^d, \quad J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega)(\theta)$$
$$= \int_{\partial\Omega} \left(j(u_\Omega) + \nabla u_\Omega \cdot \nabla p_\Omega + u_\Omega p_\Omega - f p_\Omega \right) \theta \cdot n \, \mathrm{d}s.$$

Proof:

- Since u_Ω is the solution to the state equation, it holds:

$$\text{For any } q \in H^1(\mathbb{R}^d), \quad \mathcal{L}(\Omega, u_\Omega, q) = \int_\Omega j(u_\Omega) \, \mathrm{d}x = J(\Omega).$$

- Differentiating with respect to Ω in a direction θ and using the chain rule yield:

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

where $u'_\Omega(\theta)$ is the Eulerian derivative of $\Omega \mapsto u_\Omega$ (which is assumed to exist).

Céa's method: the Neumann case (IV)

- Now, choosing $q = p_\Omega$ and using the cancellation of the adjoint

$$\forall \widehat{u} \in H^1(\mathbb{R}^d), \quad \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega)(\widehat{u}) = 0,$$

we obtain:

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

- Eventually, applying the "elementary" shape derivative formula:

$$F(\Omega) = \int_\Omega f(x)\,\mathrm{d}x \quad \Rightarrow \quad F'(\Omega)(\theta) = \int_{\partial\Omega} f\,\theta\cdot n\,\mathrm{d}s,$$

to compute the above partial shape derivative $\frac{\partial \mathcal{L}}{\partial \Omega}$ of the Lagrangian

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

we arrive at the desired formula:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \Big(j(u_\Omega) + \nabla u_\Omega\cdot\nabla p_\Omega + u_\Omega p_\Omega - f p_\Omega\Big)\,\theta\cdot n\,\mathrm{d}s.$$

Céa's method: the Dirichlet case (I)

- Let us consider the counterpart situation featuring Dirichlet boundary condition:

$$J(\Omega) = \int_\Omega j(u_\Omega)\,\mathrm{d}x, \quad \text{where} \quad \begin{cases} -\Delta u_\Omega = f & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \partial\Omega. \end{cases}$$

- **Difficulty:** Since $u = 0$ on $\partial\Omega$, the arguments of a "naive" Lagrangian are not independent.

- **Solution:** We penalize this boundary condition by adding an extra variable $\lambda \in H^1(\mathbb{R}^d)$ to the Lagrangian:

For all $u, p, \lambda \in H^1(\mathbb{R}^d)$,

$$\mathcal{L}(\Omega, u, p, \lambda) = \underbrace{\int_\Omega j(u)\,\mathrm{d}x}_{\substack{\textbf{Objective function} \\ \textbf{where } u_\Omega \textbf{ is replaced by } u}} + \underbrace{\int_\Omega (-\Delta u - f)p\,\mathrm{d}x}_{\substack{\textbf{penalization of the} \\ \text{“}\textbf{constraint}\text{”} -\Delta u = f}} + \underbrace{\int_{\partial\Omega} \lambda u\,\mathrm{d}s}_{\substack{\textbf{penalization of the} \\ \text{“}\textbf{constraint}\text{”} \ u=0 \textbf{ on } \partial\Omega}}.$$

Theorem.

The shape derivative of the objective function $J(\Omega)$ equals:

$$\text{For all } \theta : \mathbb{R}^d \to \mathbb{R}^d, \quad J'(\Omega)(\theta) = \int_{\partial\Omega} \left(j(u_\Omega) - \frac{\partial u_\Omega}{\partial n} \frac{\partial p_\Omega}{\partial n} \right) \theta \cdot n \, \mathrm{d}s$$

where the adjoint state p_Ω is the solution to:

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

Remarks.

- The shape derivative formula is different from that in the Neumann case.
- The optimal value of the extra Lagrange multiplier equals $\lambda_\Omega = -\frac{\partial p_\Omega}{\partial n}$ on $\partial\Omega$.

Céa's method: Two-phase shape optimization (I)

We optimize the interface between two phases:

$$\min_{\Omega \subset D} J(\Omega) = \int_D j(u_\Omega)\,\mathrm{d}x.$$

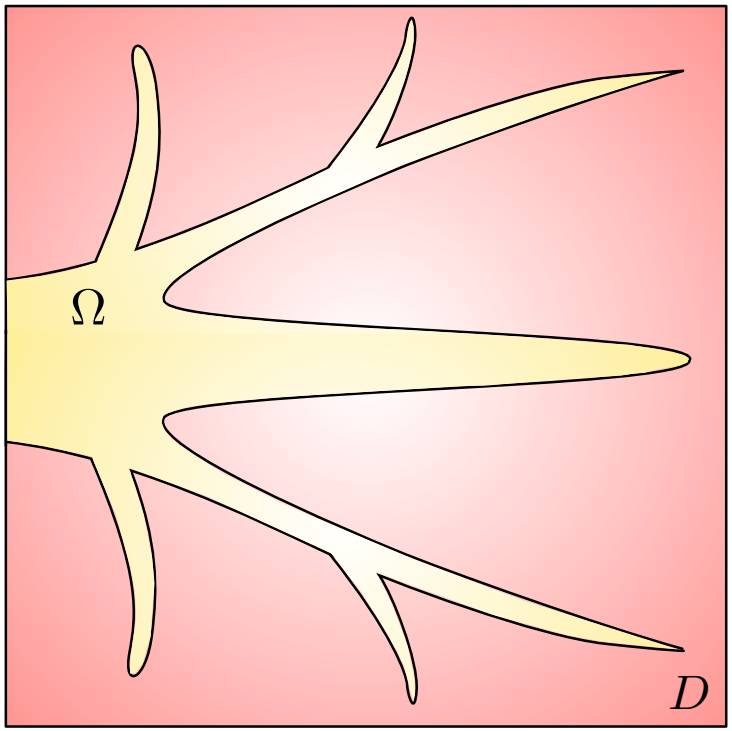
In this formulation,

- D is a fixed "hold-all domain",
- The conductivity γ_Ω inside D reads:

$$\gamma_\Omega(x) = \begin{cases} \beta & \text{if } x \in \Omega, \\ \alpha & \text{if } x \in D \setminus \Omega. \end{cases}$$

- The state $u_\Omega \in H^1(D)$ is solution to:

$$\begin{cases} -\mathrm{div}(\gamma_\Omega \nabla u_\Omega) = f & \text{in } D, \\ u_\Omega = 0 & \text{on } \partial D. \end{cases}$$



Difficulty: The gradient ∇u_Ω is discontinuous across $\partial\Omega$, since

$$u_\Omega^{\text{in}} = u_\Omega^{\text{out}} \quad \text{and} \quad \beta \frac{\partial u_\Omega^{\text{in}}}{\partial n} = \alpha \frac{\partial u_\Omega^{\text{out}}}{\partial n} \quad \text{on } \partial\Omega,$$

where u_Ω^{in} and u_Ω^{out} denote the restrictions of u_Ω to Ω and $D \setminus \overline{\Omega}$, respectively.

Céa's method: Two-phase shape optimization (II)

Remedy: We introduce two extra variables $\lambda, \mu \in H^1(D)$ in the Lagrangian to penalize the transmission conditions:

$$\mathcal{L}(\Omega, u, p, \lambda, \mu) = \int_D j(u)\,\mathrm{d}x + \int_{\Omega\cup(D\setminus\Omega)} (-\mathrm{div}(\gamma_\Omega \nabla u) - f)p\,\mathrm{d}x + \int_{\partial\Omega} \lambda[u]\mathrm{d}s + \int_{\partial\Omega} \mu\left[\gamma_\Omega \frac{\partial u}{\partial n}\right]\mathrm{d}s$$

where $u = (u^{\mathrm{in}}, u^{\mathrm{out}})$, $p = (p^{\mathrm{in}}, p^{\mathrm{out}})$ are discontinuous at $\partial\Omega$ and $[a] := a^{\mathrm{in}} - a^{\mathrm{out}}$.

Theorem.

The shape derivative of $J(\Omega)$ reads:

$$\text{For all } \theta : \mathbb{R}^d \to \mathbb{R}^d, \quad J'(\Omega)(\theta) = (\alpha - \beta)\int_{\partial\Omega} \nabla_T u_\Omega \cdot \nabla_T p_\Omega\, \theta \cdot n\,\mathrm{d}s - \left(\frac{1}{\alpha} - \frac{1}{\beta}\right)\int_{\partial\Omega} \left(\gamma_\Omega \frac{\partial u_\Omega}{\partial n}\right)\left(\gamma_\Omega \frac{\partial p_\Omega}{\partial n}\right)\theta \cdot n\,\mathrm{d}s,$$

where $\nabla_T u := \nabla u - \frac{\partial u}{\partial n} n$ is the tangential gradient and the adjoint state $p_\Omega \in H^1(\mathbb{R}^d)$ is the solution to:

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

Part II

Advance practice of the Level Set Method

The "Hilbertian trick" (I)

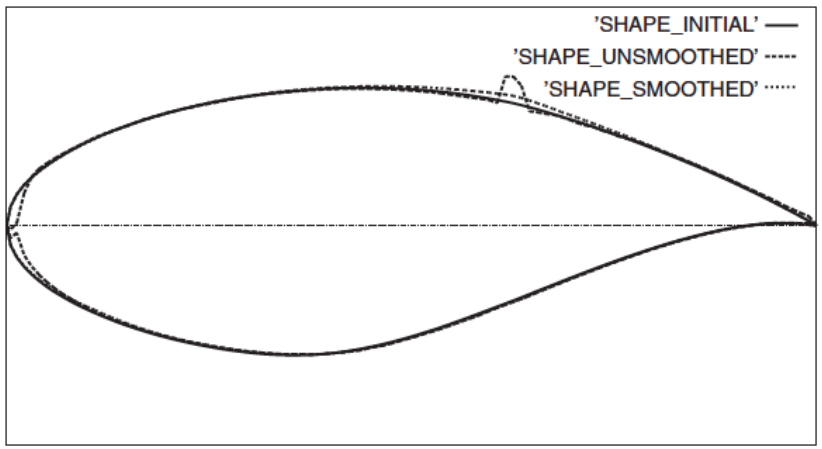
- The shape derivative of "most" functionals $J(\Omega)$ has the Hadamard structure:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} v_\Omega(\theta \cdot n)\,\mathrm{d}s, \text{ for some function } v_\Omega : \partial\Omega \to \mathbb{R}.$$

- A descent direction for $J(\Omega)$ is readily suggested by this formula:

$$\theta = -v_\Omega n \text{ on } \partial\Omega \quad \Rightarrow \quad J'(\Omega)(\theta) < 0.$$

- This "naive" choice is actually awkward for multiple reasons:
  - It solely prescribes θ on $\partial\Omega$, while in practice, it is crucial to define θ on $\mathbb{R}^d$.
  - It may be irregular, e.g. because of numerical errors in the Finite Element analyses.



The use of the irregular shape gradient in the very sensitive problem of drag minimization of an airfoil immediately causes large undesirable artifacts (Taken from [MoPir]).

The "Hilbertian trick" (II)

- The "Hilbertian trick" is a principled method to extend and regularize the shape gradient into a new field ξ which is
  - Smoothed, with a controlled length-scale of variations;
  - Defined on the whole ambient space $\mathbb{R}^d$;
  - Guaranteed to be a descent direction for $J(\Omega)$.
- The idea is part of the "shape optimization folklore" [AWu, Bu, Gou, MoPir].
- It is a common ingredient in the practice of gradient flows [Neu].

Reminder: Derivatives and gradients (I)

Definition (Fréchet derivative).

Let $(X, ||\cdot||_X)$ be a Banach space. A real-valued function $F : X \to \mathbb{R}$ is differentiable at $u \in X$ if there exists a linear, continuous mapping $F'(u) : X \to \mathbb{R}$ such that:

$$F(u+v) = F(u) + F'(u)(v) + o(||v||_X), \text{ where } \frac{o(||v||_X)}{||v||_X} \xrightarrow{v \to 0} 0.$$

The linear mapping $F'(u) \in X^*$ is the Fréchet derivative of F at u.

Definition (Gradient).

If in addition X contains a Hilbert space $(H, \langle \cdot, \cdot \rangle_H)$, $H \subset X$, the Riesz representation theorem allows to identify the derivative $F'(u)$ with an element $\nabla F(u) \in H$:

$$\forall v \in H, \quad F'(u)(v) = \langle \nabla F(u), v \rangle_H;$$

The element $\nabla F(u) \in H$ is called the gradient of F at u (with respect to H).

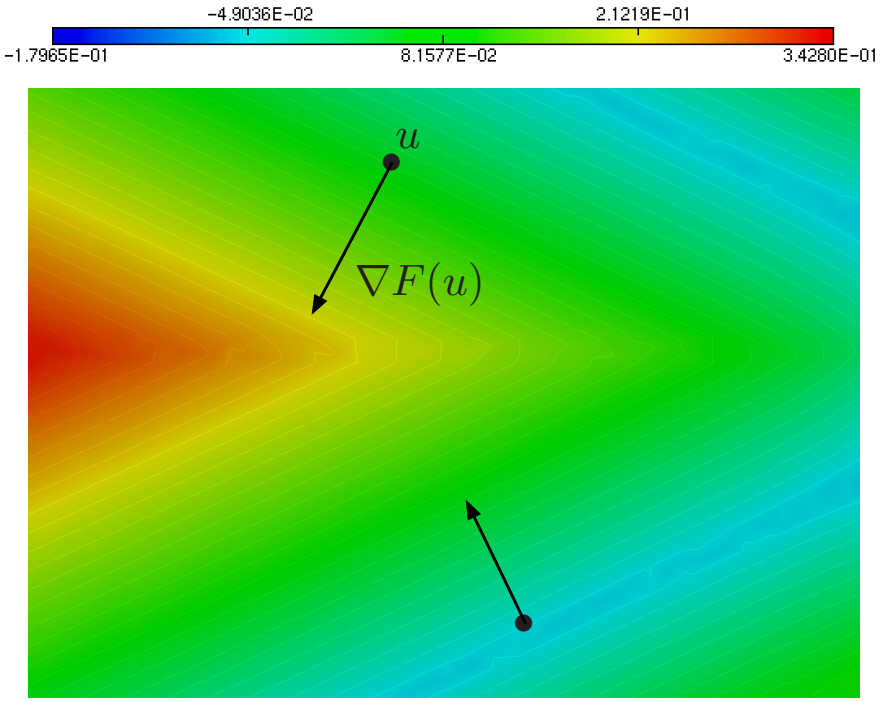
Reminder: Derivatives and gradients (II)

Interpretation: If F is differentiable at $u \in H$, then, for "small" $\tau > 0$:

$$\begin{array}{rcl} \forall \widehat{u} \in H,\ ||\widehat{u}||_H \leq 1, \quad F(u - \tau\widehat{u}) & \approx & F(u) - \tau\langle \nabla F(u), \widehat{u} \rangle_H, \\ & \geq & F(u) - \tau||\nabla F(u)||_H, \end{array}$$

where equality holds if and only if $\widehat{u} = \frac{\nabla F(u)}{||\nabla F(u)||_H}$ (Cauchy-Schwarz inequality).

$\Rightarrow -\nabla F(u)$ is the best descent direction for F from u.



Some isolines of a function $F : \mathbb{R}^2 \to \mathbb{R}$ and the gradient $\nabla F(u) \in \mathbb{R}^2$ at some point $u \in \mathbb{R}^2$.

The Hilbertian method (I)

- The "naive" (scalar) descent direction $-v_\Omega$ stems from the formula:

$$J'(\Omega)(\theta) = \underbrace{\int_{\partial\Omega} v_\Omega\, \theta\cdot n \,\mathrm{d}s}_{=:\langle v_\Omega, \theta\cdot n\rangle_{L^2(\partial\Omega)}} \qquad \forall \theta \in \mathcal{C}^1(\mathbb{R}^d;\mathbb{R}^d),$$

  i.e. v_Ω is the $L^2(\partial\Omega)$ gradient of $J'(\Omega)$, but $L^2(\partial\Omega)$ is not a subset of $\mathcal{C}^1(\mathbb{R}^d)$.

- **Idea:** Trade the "natural" $L^2(\partial\Omega)$ inner product for that $a(\cdot,\cdot)$ on a Hilbert space H of "more regular" functions on D:

> Search for the scalar field $g \in H$ s.t. $\forall w \in H, \quad a(g,w) = J'(\Omega)(wn)$.

- The vector field $-gn$ is still a descent direction for $J(\Omega)$, since

$$0 < a(g,g) = J'(\Omega)(gn) \quad \text{for the choice } w = g \neq 0.$$

  Thus, for "small" $\tau > 0$

$$J((\mathrm{Id} - \tau gn)(\Omega)) = J(\Omega) - \tau a(g,g) + o(\tau) < J(\Omega).$$

- It is naturally defined on D, and inherits the regularity of functions in H.

The Hilbertian method (II)

- Introducing a large computational domain D, a common choice in practice is:

$$H = H^1(D), \text{ with inner product } a(u,v) = \alpha^2 \int_D \nabla u \cdot \nabla v \, \mathrm{d}x + \int_D uv \, \mathrm{d}x.$$

- The identification problem for $g \in H^1(D)$:

$$\forall w \in H^1(D), \quad \alpha^2 \int_D \nabla g \cdot \nabla w \, \mathrm{d}x + \int_D gw \, \mathrm{d}x = J'(wn)$$

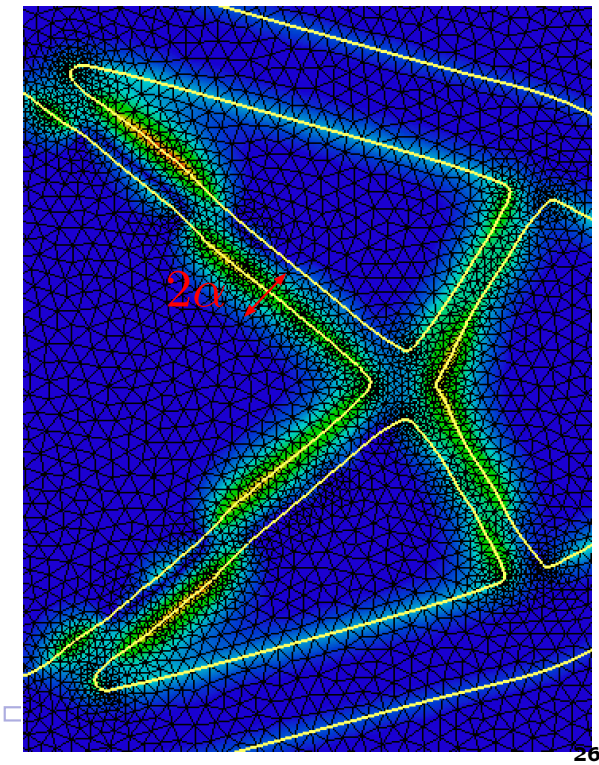
is a simple Laplace-like variational problem.

- The regularized field g writes:

$$g(x) = \int_{\partial\Omega} G\left(\frac{x-y}{\alpha}\right) v_\Omega(y) \, \mathrm{d}s,$$

where G is the fundamental solution of $u \mapsto -\Delta u + u$.

- The parameter α is a regularization length scale for the information in $J'(\Omega)$.
- In practice, $\alpha \approx$ a few mesh sizes.
- This idea is formal: $H^1(D)$ is not a subset of $\mathcal{C}^1(D)$.



The Hilbertian method (III)

- Multiple other regularizing problems are possible, associated to different inner products or different function spaces, allowing for instance to:
  - Extend the vector velocity $\theta = -v_\Omega n$, by using an inner product over vector fields such as that of linear elasticity.
  - Require the descent direction to vanish on imposed or non design regions of D.

- This Hilbertian method also allows to extract a descent direction from the volume form of the shape derivative:

$$J'(\Omega)(\theta) = \int_\Omega (r_\Omega \cdot \theta + S_\Omega : \nabla\theta)\,\mathrm{d}x,$$

  with known fields $r_\Omega : \Omega \to \mathbb{R}^d$, $S_\Omega : \Omega \to \mathbb{R}^{d\times d}$, see [GiaPanTra] about this issue.

- It also lends itself to the deployment of constrained optimization algorithms...

Part II

Advance practice of the Level Set Method

Constrained optimization algorithms

Let us consider a generic constrained optimization problem:

$$\min_{h \in H} J(h) \text{ s.t. } G(h) = 0. \qquad (\mathcal{P})$$

In this formulation,

- The design variable h belongs to a Hilbert space H with inner product $a(\cdot, \cdot)$.
- $J(h)$ is an objective function.
- $G(h) = (G_1(h), \ldots, G_p(h))$ is a collection of p (equality) constraints.

Problems of the form $(\mathcal{P})$ can be treated by a variety of methods:

- The augmented Lagrangian method, Sequentiel Linear Programming... [NoWri].
- We present the null-space algorithm [FeAlDa], based on linearization of $(\mathcal{P})$.

The null-space algorithm (I)

The resolution of $(\mathcal{P})$ proceeds by iterations of the form:

$$h^{n+1} = h^n + \tau^n \xi^n, \text{ where } \xi^n := -\left(\alpha_J \xi_J^n + \alpha_G \xi_G^n\right).$$

This update procedure is made of:

- A null-space direction ξ_J^n which best minimizes J without altering constraints;
- A range-space direction ξ_G^n aimed to decrease the violation of constraints;
- Weights α_J and α_G that encode the relative priority of minimizing $J(h)$ and fulfilling the constraints;
- A sufficiently small (pseudo)-time step τ^n.

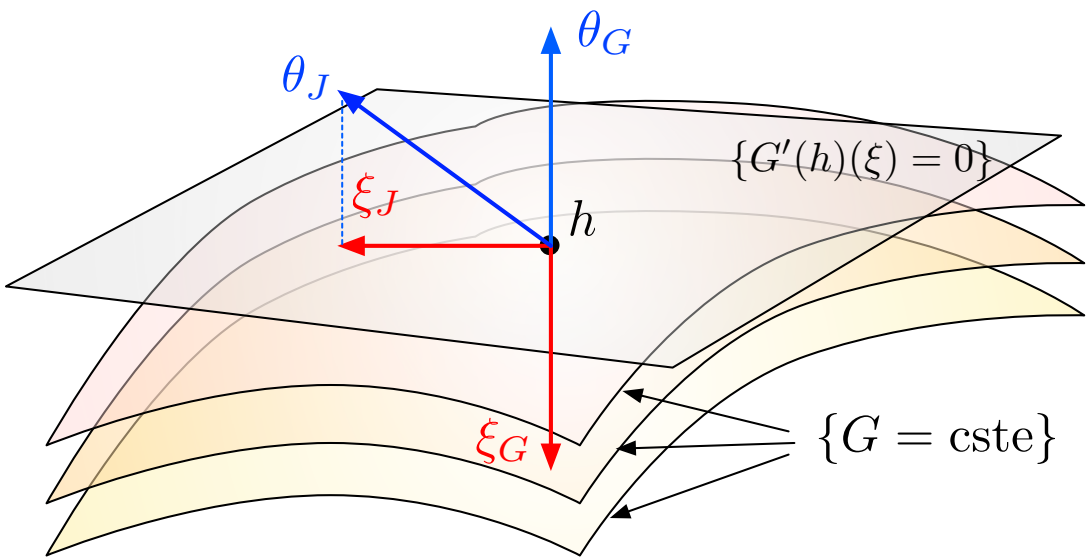
The null-space algorithm (II)

- Let us place ourselves at a given iteration $h = h^n$ of the process.
- We denote by θ_J and θ_{G_i} the gradients of $J(h)$ and $G_i(h)$ at h:

$$\forall \xi \in H, \quad a(\theta_J, \xi) = J'(h)(\xi), \text{ and } \forall \xi \in H, \quad a(\theta_{G_i}, \xi) = G_i'(h)(\xi).$$

- We search for $\xi_J \perp \xi_G$ as linear combinations of θ_J and θ_{G_i}:

$$\xi_J = \theta_J - \sum_{i=1}^{p} \lambda_i \theta_{G_i}, \text{ and } \xi_G = \sum_{i=1}^{p} \beta_i \theta_{G_i}$$



The null-space algorithm (III): null-space direction

- The null-space direction ξ_J is the orthogonal projection of θ_J on the null-space of constraints:
$$G_i'(h)(\xi_J) = a(\theta_{G_i}, \xi_J) = 0, \text{ for } i = 1, \ldots, p.$$

- **Interpretation:** $-\xi_J$ is the "best" descent direction for $J(h)$ that leaves the constraint functional $G(h)$ unchanged at first order.

- Using elementary algebra, this characterizes $\boldsymbol{\lambda} \in \mathbb{R}^p$ in $\xi_J = \theta_J - \sum_{i=1}^{p} \lambda_i \theta_{G_i}$ as the solution to:
$$S\boldsymbol{\lambda} = \mathbf{b},$$
where
  - $S \in \mathbb{R}^{p \times p}$ is the matrix with entries $S_{ij} = a(\theta_{G_i}, \theta_{G_j})$, which is invertible under a classical qualification condition on the constraints (LICQ = Linear Independence Constraint Qualification),
  - $\mathbf{b} \in \mathbb{R}^p$ is the vector with components $b_i = a(\theta_J, \theta_{G_i})$.

The null-space algorithm (IV): range-space direction

- The range space direction ξ_G guarantees that

$$G_i(h - \tau\alpha_G\xi_G) = (1 - \alpha_G\tau)G_i(h) + o(\tau), \text{ for } i = 1, \ldots, p,$$

  i.e. moving along $-\alpha_G\xi_G$ reduces the constraints by a rate α_G.

- Using a first-order expansion, this rewrites:

$$G'(h)(\xi_G) \approx G(h).$$

- This characterizes the coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$, such that $\xi_G = \sum_{i=1}^{p} \beta_i \theta_{G_i}$, as:

$$S\boldsymbol{\beta} = \mathbf{c},$$

  where

  - $S \in \mathbb{R}^{p \times p}$ is again the matrix with entries $S_{ij} = a(\theta_{G_i}, \theta_{G_j})$;
  - $\mathbf{c} \in \mathbb{R}^p$ is the vector with components $c_i = G_i(h)$.

The null-space algorithm (V): final comments

- At each iteration n, a merit function is used to check the step size τ^n:

$$M^n(h) := \alpha_J\Big(J(h) - \boldsymbol{\lambda}^n \cdot G(h)\Big) + \frac{\alpha_G}{2}(S^n)^{-1}G(h)\cdot G(h),$$

  i.e τ^n is deemed "too large" if

$$M^n(h^n + \tau^n\xi^n) > M^n(h^n).$$

- The choice of this merit function is motivated by the fact that

$$(M^n)'(h^n) = \alpha_J\xi_J + \alpha_G(S^n)^{-1}G(h^n)\cdot G'(h^n) \equiv \xi^n$$

- This strategy is almost parameter-free: it solely relies on 2 meaningful parameters:

  α_J, $\alpha_G \approx$ relative decrease rates of the objective and the violation of constraints.

- This method also accommodates inequality constraints very efficiently.

- With minor adaptations, it can be deployed for shape optimization thanks to the Hilbertian method, as any linearization-based optimization algorithm.

Numerical example (I)

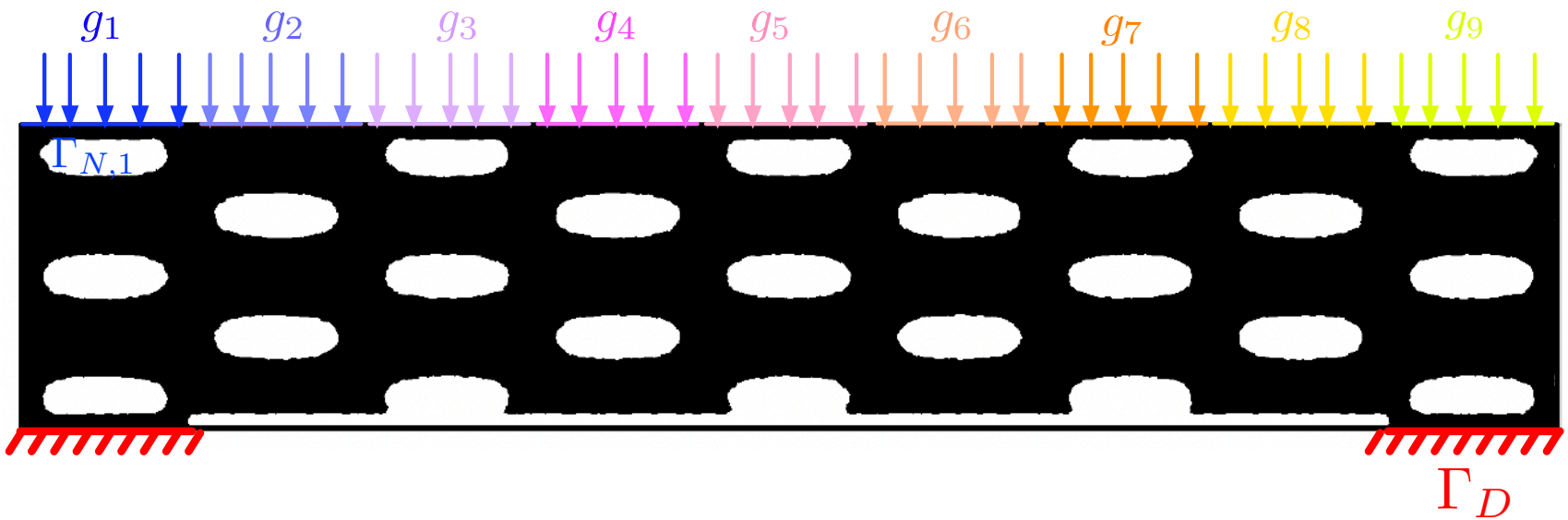
We optimize the volume of a 2d bridge under multiple compliance constraints:

$$\min_{\Omega} \mathrm{Vol}(\Omega) \text{ s.t. } C_i(\Omega) \leq C_{\mathsf{T}},$$

where

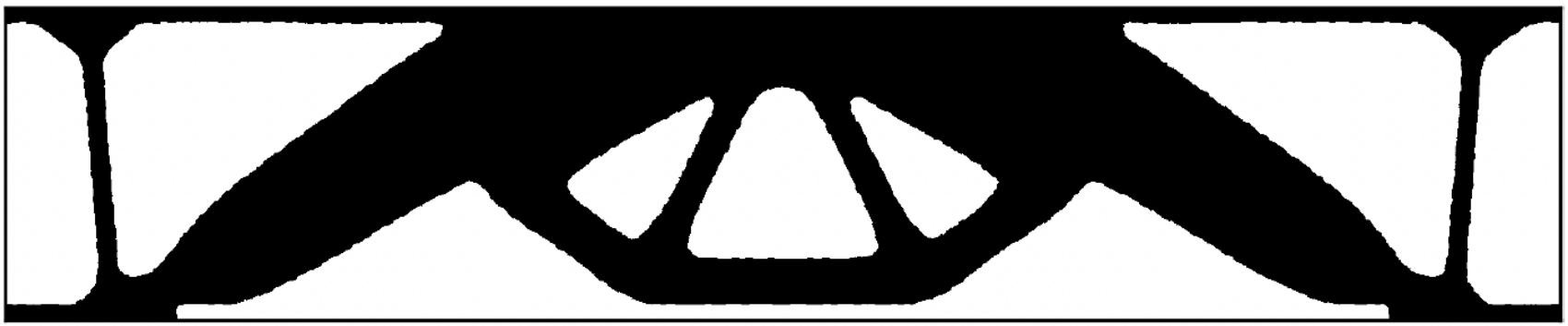
$$C_i(\Omega) = \int_{\Gamma_{N,i}} g_i \cdot u_{\Omega,i} \, \mathrm{d}s$$

is the compliance of Ω when it is submitted to the load g_i on $\Gamma_{N,i} \subset \partial D$.

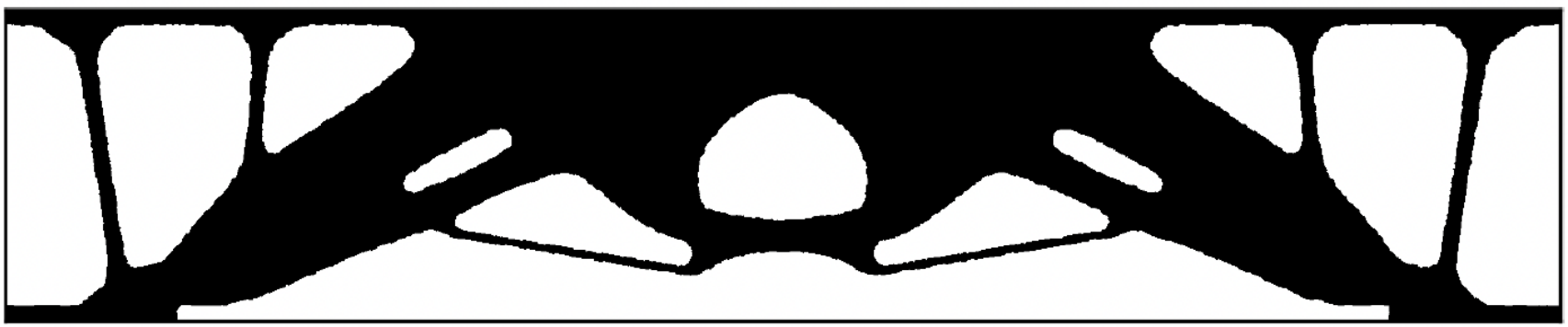


Numerical example (II)

Optimized bridge under application of 1 *load.*



Optimized bridge under application 3 *loads.*

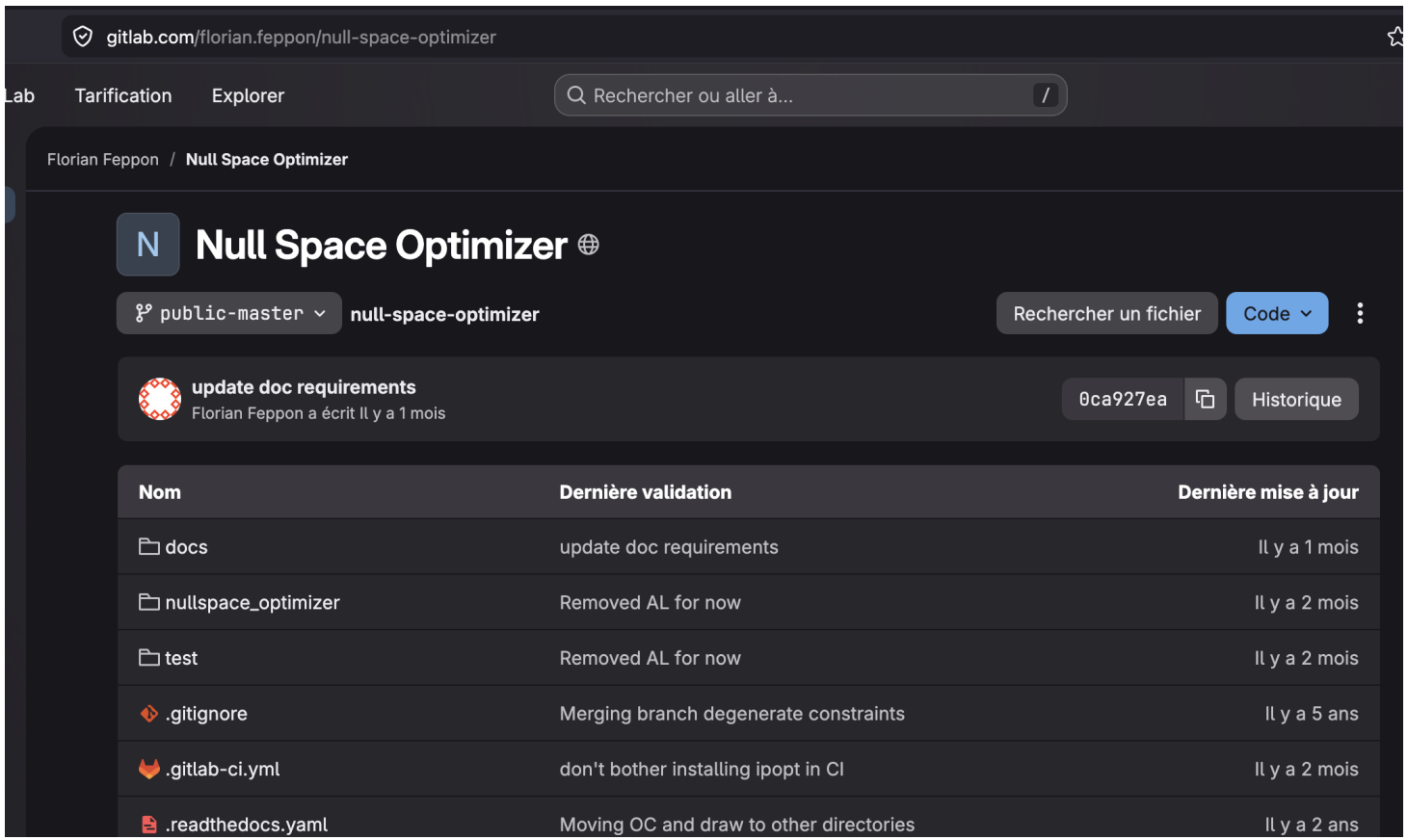


Optimized bridge under application of all 9 *loads.*

Open-source implementation

`null-space optimizer` [FeAlDa]: A `python`-based implementation, by F. Feppon.

`https://gitlab.com/florian.feppon/null-space-optimizer`



Part II

Advance practice of the Level Set Method

Reminder of Webinar 1 (I)

- Let $\Omega(t) \subset \mathbb{R}^d$ be a domain evolving according to a velocity field $V(t,x) \in \mathbb{R}^d$.

  In shape optimization, $V(t,x)$ is a descent direction.

- Let $\phi(t,x)$ be a Level Set function for $\Omega(t)$.

- The motion of $\Omega(t)$ translates in terms of ϕ as the "advection-like" equation:

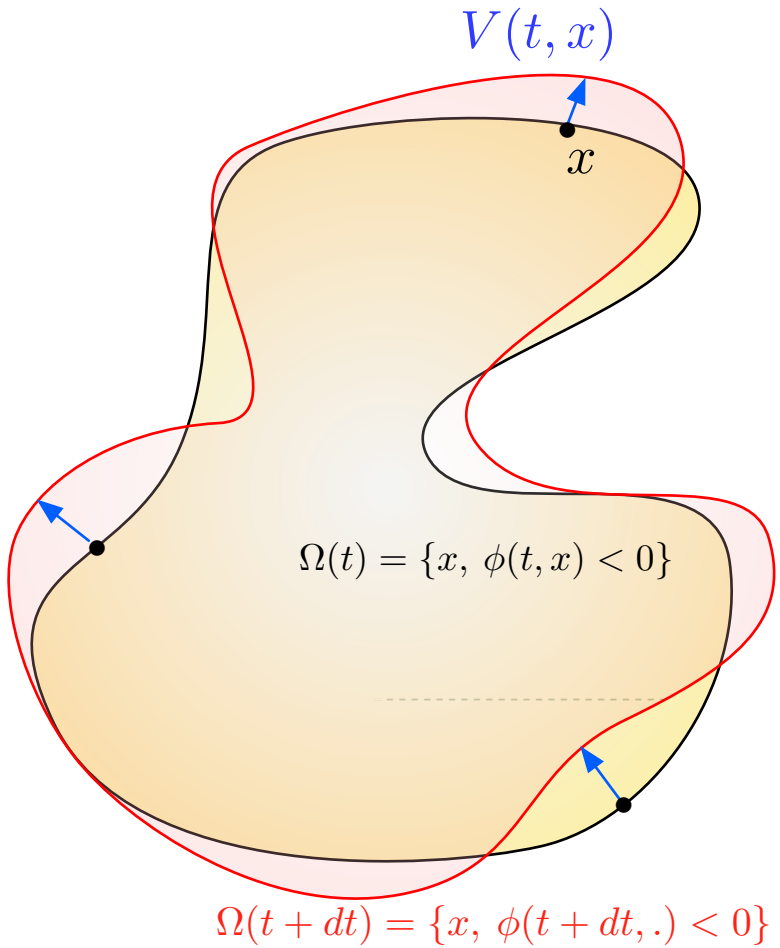
$$\boxed{\frac{\partial \phi}{\partial t}(t,x) + V(t,x) \cdot \nabla \phi(t,x) = 0}$$

- The motion depends only on the normal part

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

and this rewrites as a Hamilton-Jacobi equation:

$$\boxed{\frac{\partial \phi}{\partial t}(t,x) + v(t,x)|\nabla \phi(t,x)| = 0.}$$



Reminder of Webinar 1 (II)

- In practice, the time interval $(0, T)$ is decomposed into a series of sub-intervals:

$$0 = t^0 < t^1 < \ldots < t^N = T,$$

and the normal velocity $v(t, x)$ is frozen on each of them:

$$v(t, x) \approx v^n(x) := v(t^n, x) \text{ for } t \in (t^n, t^{n+1}).$$

- Doing so yields a Hamilton-Jacobi equation with steady-state velocity $v(x)$:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t, x) + v(x)|\nabla \phi(t, x)| = 0 & \text{for } (t, x) \in (0, T) \times \mathbb{R}^d, \\ \phi(t = 0, x) = \phi_0(x) & \text{for } x \in \mathbb{R}^d. \end{cases}$$

- The latter can be solver efficiently with the Osher-Sethian algorithm when the computational domain is equipped with a Cartesian grid.
- The theory of schemes for Hamilton-Jacobi equations can be adapted to the context of a simplicial mesh $\mathcal{T}$, but it is a difficult task [Abgrall] [BaSe].

The Level Set advection equation (I)

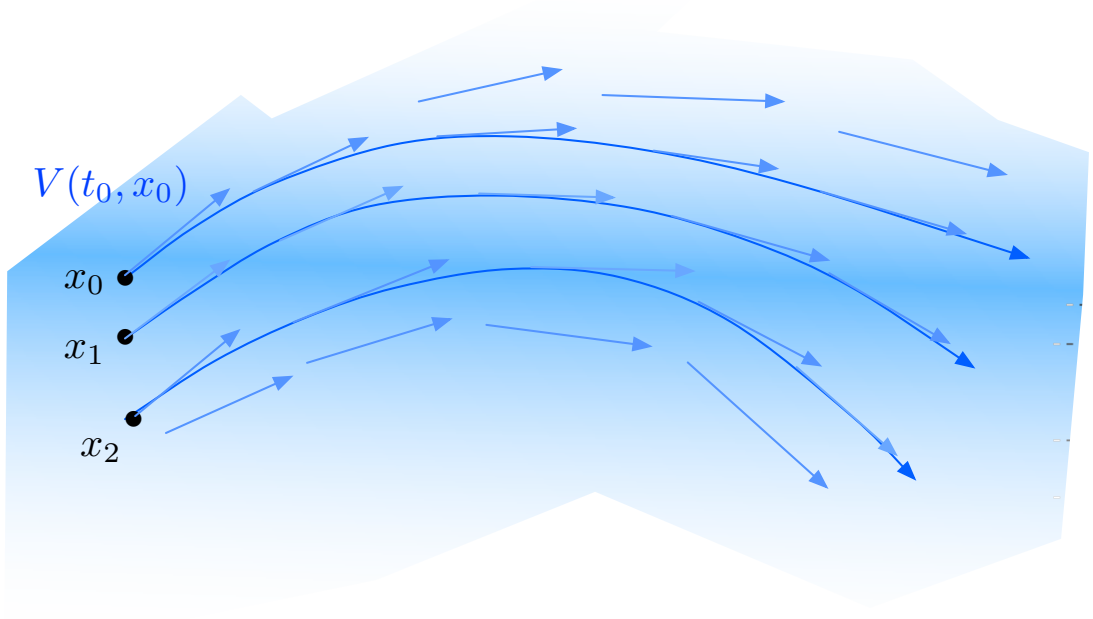
- An alternative practice is to freeze the whole vector field $V(t,x)$ over each sub-interval:
$$V(x) \approx V^n(x) := V(t^n, x) \text{ for } t \in (t^n, t^{n+1}).$$
- This yields an advection equation over a generic time period $(0,T) = (t^n, t^{n+1})$:
$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t,x) + V(x) \cdot \nabla \phi(t,x) = 0 & \text{on } (0,T) \times \mathbb{R}^d, \\ \phi(0,.) = \phi_0 & \text{on } \mathbb{R}^d, \end{cases} \quad \text{(ADV)}$$
for given velocity field $V(x)$ $(= V^n(x))$, and initial function ϕ_0 $(= \phi(t^n, \cdot))$.
- Efficient numerical schemes exist for this linear equation.
- This comes at the price of a loss of information: the consistent normal orientation of $V(t,x)$ is not retained.
- We present a strategy based on the method of characteristics [Pironneau, Strain] (explicit scheme, yet unconditionally stable).

The Level Set advection equation (II)

Definition (Characteristic curve).

Let $V : \mathbb{R}_t \times \mathbb{R}^d_x \to \mathbb{R}^d$ *be a smooth velocity field. The characteristic curve* $t \mapsto \chi(x, t, t0)$ *emerging from a point* $x \in \mathbb{R}^d$ *at time* $t = t_0$ *is defined by the ODE:*

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}(\chi(x, t, t_0)) = V(t, \chi(x, t, t_0)) & \text{for } t \in (0, T), \\ \chi(x, t_0, t_0) = x. \end{cases}$$



Three characteristic curves of the velocity field V *starting at* $t = t_0$ *from different points* x_0, x_1, x_2.

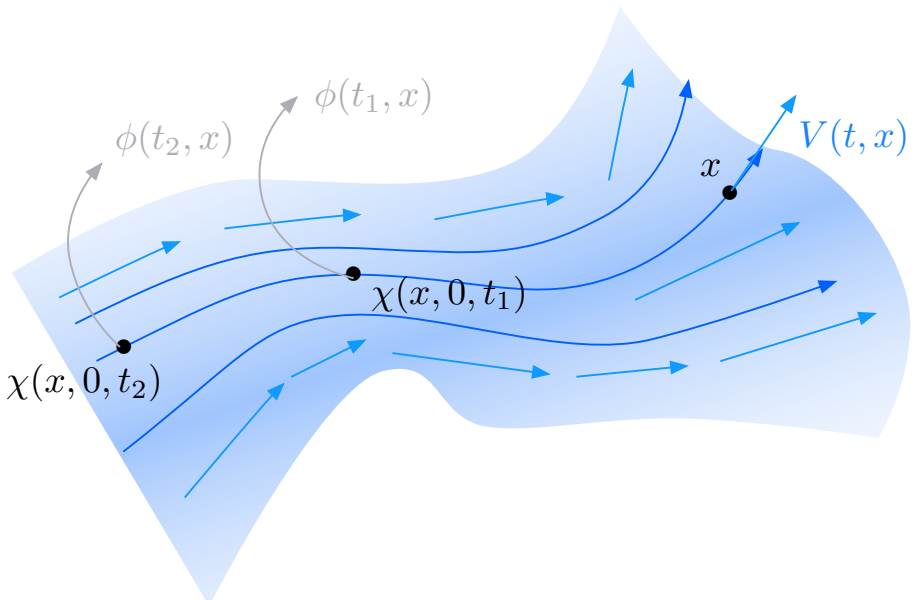
The Level Set advection equation (III)

Theorem.

The exact solution $\phi(t,x)$ to the advection equation (ADV) is:

$$\underbrace{\phi(t,x)}_{\text{Value of } \phi \text{ at } (t,x)} = \underbrace{\phi(0,\chi(x,0,t))}_{\substack{\text{Initial value of } \phi \\ \text{at the initial position of the particle at } x \text{ at } t}} ,$$

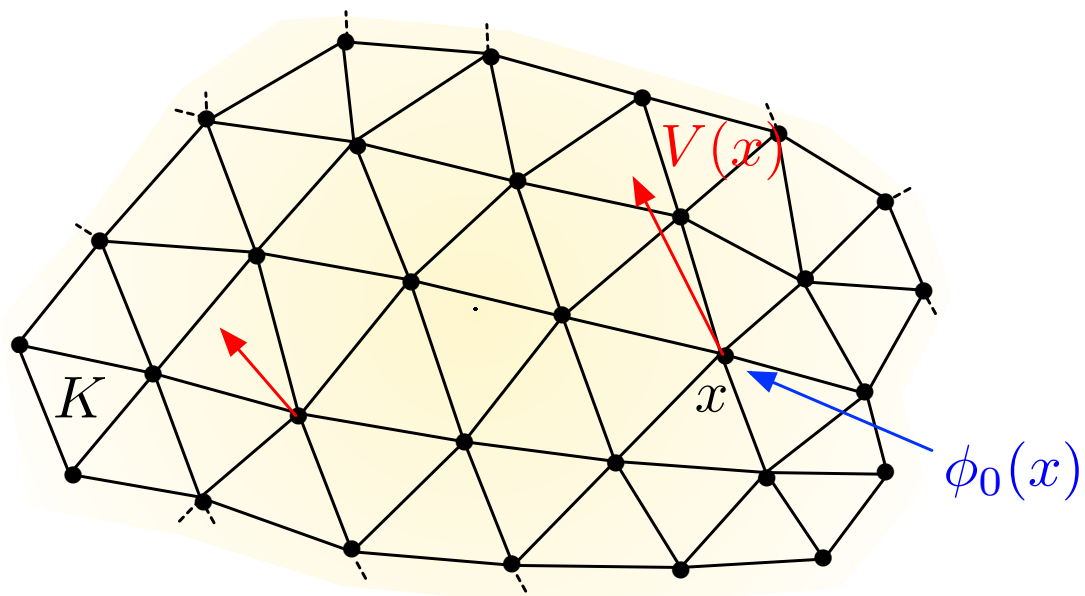
where $t \mapsto \chi(x,0,t)$ is the foot of the characteristic passing through x at time t.



Solution of the advection equation by the backward characteristic method.

Numerical resolution of the advection equation (I)

- For simplicity, we consider the solution of the advection equation in 2d.
- Let $t^n = n\Delta t$ be a discretization of the time interval $(0, T)$.
- The computational domain D is equipped with a triangular mesh $\mathcal{T}$.
- The initial datum ϕ_0 and the velocity field V are discretized at the vertices of $\mathcal{T}$.
- They are linearly interpolated from these values when needed elsewhere.

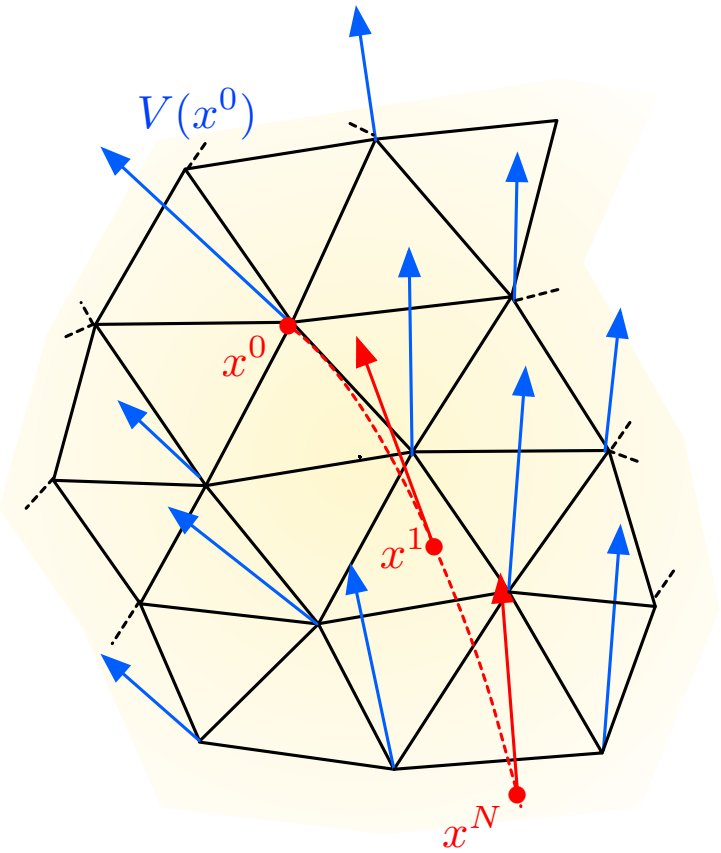


- We calculate the solution values $\phi(T, x)$ at the vertices $x \in \mathcal{T}$.

Numerical resolution of the advection equation (II)

A simple implementation hinges on Euler's integration of the characteristic curves:

- **Initialization:** Level Set function ϕ_0 at the vertices of $\mathcal{T}$.
- **For all vertices** $x \in \mathcal{T}$**:**
  - Set $x^0 = x$;
  - **For** $n = 0, \ldots, N-1$**:**
    1. Find $K^n \in \mathcal{T}$ such that $x^n \in K^n$;
    2. Calculate V at x^n by interpolation from its values at the vertices of K;
    3. $x^{n+1} = x^n - \Delta t V(x^n)$.
  - $\phi(T, x) = \phi_0(x^N)$.

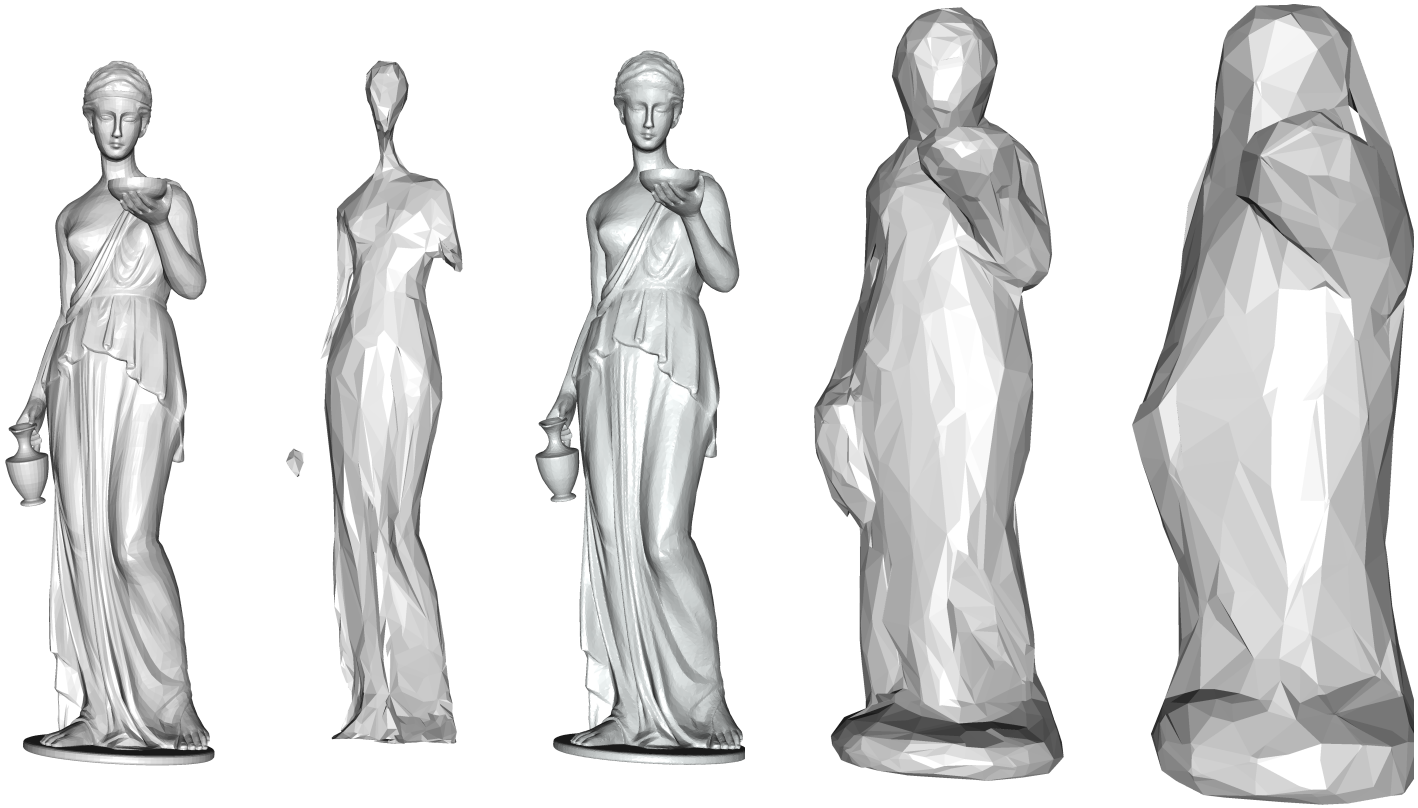


Remarks:

- This strategy is cheap: ODEs are solved explicitly.
- Its efficiency can be improved by a Runge-Kutta scheme.

The Fast Marching Method on a simplicial mesh.

The Fast Marching algorithm can also be adapted to a simplicial mesh [KiSe].

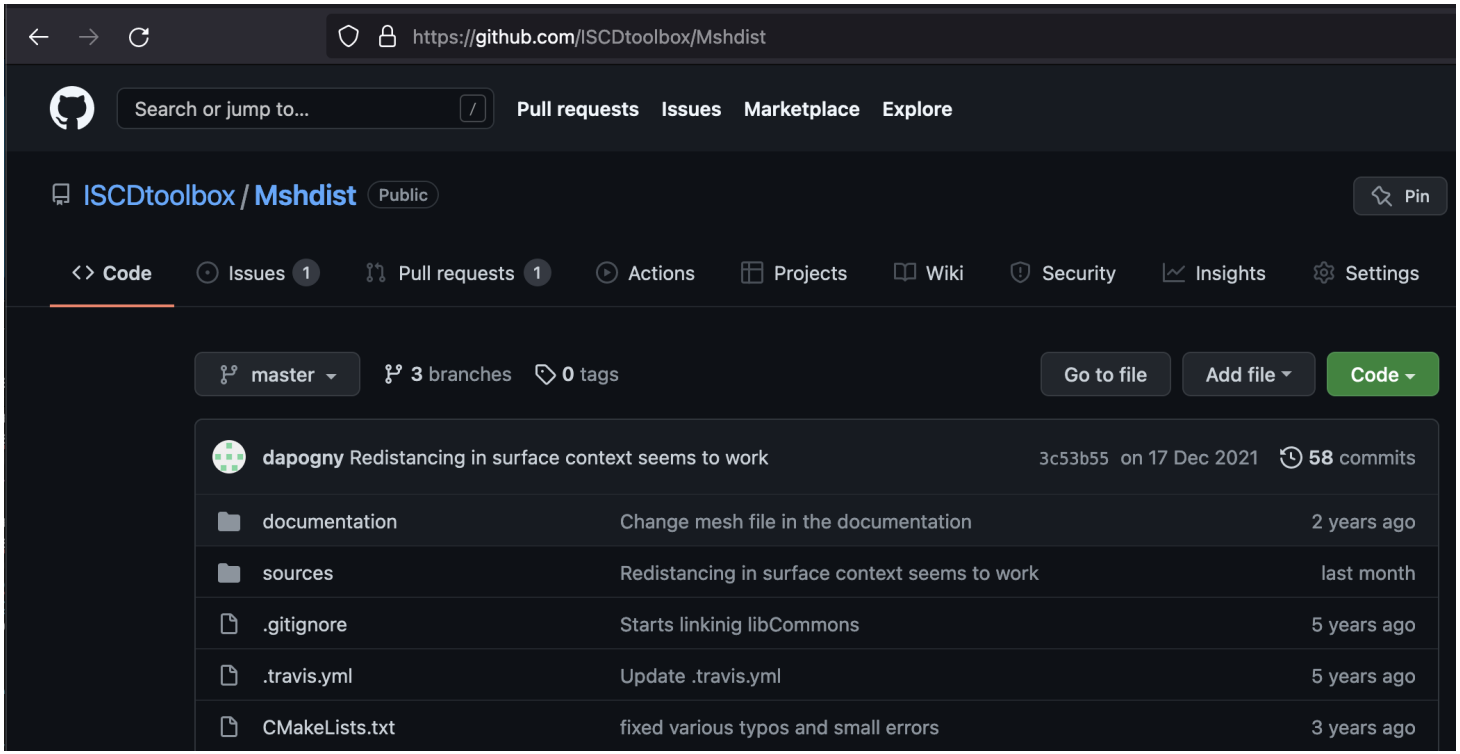


Isosurfaces of the signed distance function to the "Aphrodite" in (a): (b): isosurface -0.01, *(c): isosurface* 0, *(d): isosurface* 0.02, *(e): isosurface* 0.05.

A word of advertising: open-source implementations

- `mshdist` [DaFre]: Calculation of the signed distance function on simplicial meshes.

`https://github.com/ISCDtoolbox/Mshdist`



- `advect` [BuDaFre]: Resolution of the level set equations on simplicial meshes.

`https://github.com/ISCDtoolbox/Advect`

Part II

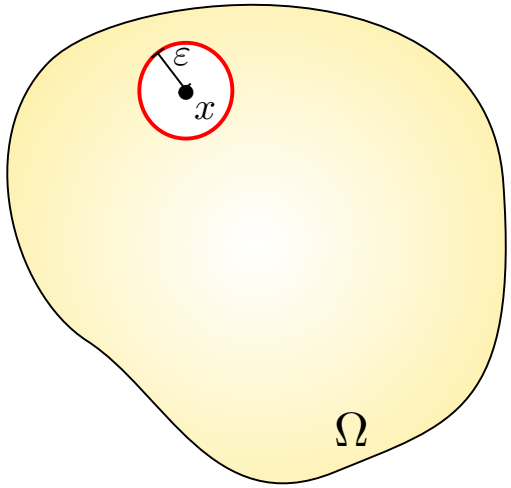
Advance practice of the Level Set Method

Differentiation with respect to the domain: topological derivatives

The notion of topological derivative features variations of a shape $\Omega \subset \mathbb{R}^d$ of the form

$$\Omega_{x,\varepsilon} := \Omega \setminus \overline{B(x,\varepsilon)},$$

where $x \in \Omega$, and $\varepsilon \ll 1$. We consider Neumann boundary conditions on $\partial B(x,\varepsilon)$.



Definition (Topological derivative).

The function $J(\Omega)$ has a topological derivative at Ω and at the point $x \in \Omega$ if there exists $\mathrm{d}_T J(\Omega)(x) \in \mathbb{R}$ such that:

$$J(\Omega_{x,\varepsilon}) = J(\Omega) + \varepsilon^d \mathrm{d}_T J(\Omega)(x) + \mathrm{o}(\varepsilon^d).$$

Intuition: If $\mathrm{d}_T J(\Omega)(x) < 0$, it is beneficial to drill a tiny hole around x.

A. A. Novotny and J. Sokołowski, *Topological derivatives in shape optimization*, Springer Science & Business Media, 2012.

Topological derivatives in linear elasticity

Example: Let u_Ω be the displacement of an elastic structure Ω,

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

and let $C(\Omega)$ denote its compliance:

$$C(\Omega) = \int_\Omega g \cdot u_\Omega \, \mathrm{d}s.$$

Theorem.

For any point $x \in \Omega$, the topological derivative of $C(\Omega)$ in 2-d equals:

$$\mathrm{d}_T C(\Omega)(x) = \frac{\pi(\lambda+2\mu)}{2\mu(\lambda+\mu)} \Big(4\mu Ae(u_\Omega) : e(u_\Omega) + (\lambda-\mu)\mathrm{tr}(Ae(u_\Omega))\mathrm{tr}(e(u_\Omega)) \Big)(x).$$

Remark: A similar formula is available in 3d.

Algorithmic deployment

- **Input:**
  - Mesh (e.g. Cartesian grid) $\mathcal{T}$ of the computational domain D;
  - Level set function ϕ^0 for the initial shape Ω^0, defined at the vertices of $\mathcal{T}$.
- **For** $n = 0, ...,$ **until convergence**
  1. Solve the state equation for u_{Ω^n} on $\mathcal{T}$.
  2. Solve the adjoint equation for p_{Ω^n} on $\mathcal{T}$.
  3. **If** n mod. $n_{\text{top}} = 0$,
     - Calculate $\mathrm{d}_T J(\Omega^n)(x)$ at all points $x \in \Omega^n$,
     - Find the point $x^n \in \Omega^n$ where $\mathrm{d}_T J(\Omega^n)(x)$ is minimum,
     - Set $\phi^{n+1}(x) = \max(\phi^n(x), \tau - |x - x^n|)$ for τ^n small enough.

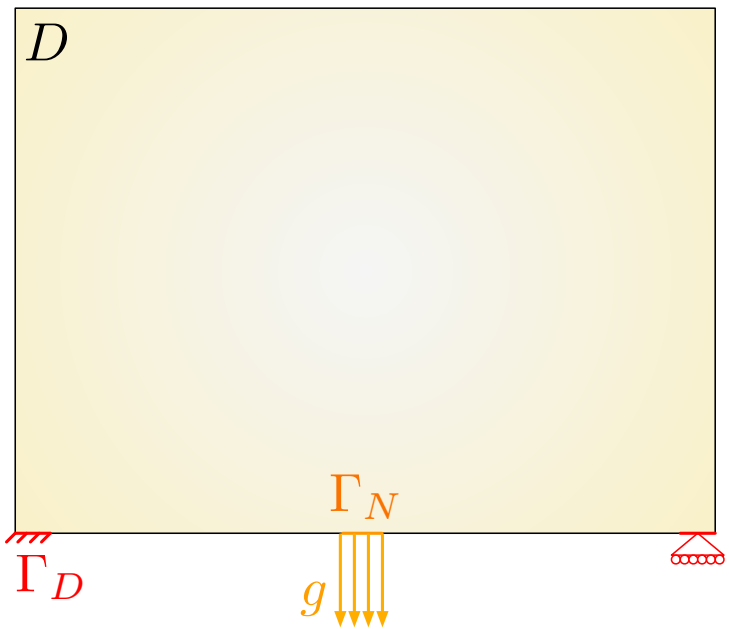
     Else: Proceed with shape derivatives.
  4. Stopping criterion.
- **Return:** Level set function ϕ^* for the optimized shape Ω^*.

Numerical example (I)

- We minimize the compliance of a bridge under a volume constraint:

$$\min_{\Omega} \; C(\Omega) \text{ s.t. } \mathrm{Vol}(\Omega) = V_{\mathsf{T}}, \text{ where } C(\Omega) = \int_{\Gamma_N} g \cdot u_\Omega \, \mathrm{d}s.$$

- The initial shape has a "simple" topology.
- We periodically try and insert holes by leveraging topological derivatives.



Numerical example (II)

Part II

A glimpse of related techniques

Other uses of the Level Set Method: a word of caution

Recommandation

Use Hadamard's shape derivatives to optimize with respect to the shape Ω.

Some authors rather use the Level Set function ϕ as the optimization variable.

1. A regularized Heaviside function is used to define material properties and integrals on the shape:

$$H_\varepsilon(\phi) = \frac{1}{2}\left(1 - \frac{\phi}{\sqrt{\phi^2 + \varepsilon^2}}\right),$$

   where $\varepsilon > 0$ is a "small" parameter.

2. Derivatives are computed with respect to ϕ, like in density methods.

3. This heavily relies on the redistanciation of ϕ into a signed distance function.

Danger of this practice

- The values of ϕ in the range $(-\varepsilon, +\varepsilon)$ are favored by the optimization algorithm because intermediate material properties are often "beneficial".
- Penalization is required (by re-distancing) but it is known to be a delicate issue.

A glimpse of related techniques

- We outline 3 popular shape and topology optimization frameworks, related to, but different from the "classical" Level Set Method.
- To set ideas, we consider a two-phase optimization problem within a domain D:

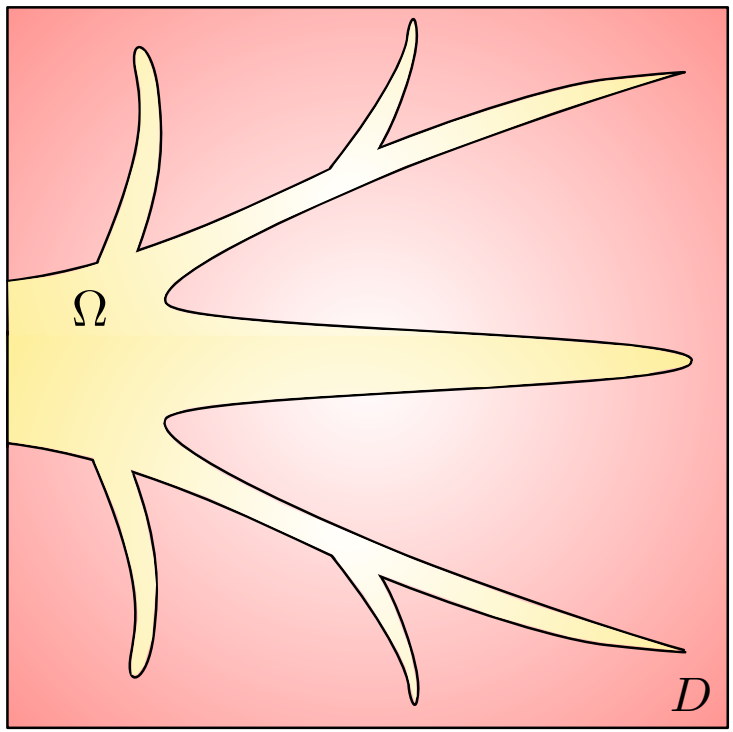
$$\min_{\Omega \subset D} J(\Omega), \text{ where } J(\Omega) = \int_D j(u_\Omega)\,\mathrm{d}x. \qquad (\mathcal{P})$$

- The state $u_\Omega \in H^1(D)$ is solution to:

$$\begin{cases} -\mathrm{div}(\gamma_\Omega \nabla u_\Omega) = f & \text{in } D, \\ u_\Omega = 0 & \text{on } \partial D, \end{cases}$$

where the conductivity γ_Ω inside D reads:

$$\gamma_\Omega(x) = \begin{cases} \beta & \text{if } x \in \Omega, \\ \alpha & \text{if } x \in D \setminus \Omega. \end{cases}$$



- One-phase and void optimization is formally retrieved at the limit $\alpha \to 0$.

Phase-field methods [BouCha, BlanGar]

- The phase-field method considers the perimeter penalized version of the problem:

$$\min_{\Omega} \; J(\Omega) + \ell \mathrm{Per}(\Omega).$$

- The repartition of materials α and β is accounted for by a phase-field function:

$$\varphi : D \to [0,1] \text{ s.t. } \begin{cases} \varphi(x) = 0 \text{ and } 1 & \text{in the pure phases } \alpha \text{ and } \beta, \\ \varphi(x) \in (0,1) & \text{where there is a mixture of both materials.} \end{cases}$$

- A material law $A(\varphi)$ assigns a conductivity to intermediate regions:

$$\begin{cases} -\mathrm{div}(A(\varphi)\nabla u_\varphi) = f & \text{in } D, \\ u_\varphi = 0 & \text{on } \partial D. \end{cases}$$

- The perimeter functional $\mathrm{Per}(\Omega)$ is approximated by:

$$F_\varepsilon(\varphi) = \underbrace{\varepsilon \int_D |\nabla \varphi|^2 \, \mathrm{d}x}_{\text{Penalization of the transitions between phases}} + \underbrace{\frac{1}{\varepsilon} \int_D W(\varphi) \, \mathrm{d}x}_{\text{Penalization of mixtures}},$$

  where $W(\cdot) \geq 0$ is a double-well potential, which equals 0 at $\varphi = 0$ or 1 only.

- The new problem

$$\min_{\varphi : D \to [0,1]} \int_D j(u_\varphi) \, \mathrm{d}x + \ell F_\varepsilon(\varphi),$$

  is a parametric optimization problem.

S. Amstutz' algorithm [AmHan, Am] (I)

- This algorithm represents the shape by a Level Set function, but appraises the sensitivity of $J(\Omega)$ with topological derivatives only.
- The topological expansion of $J(\Omega)$ is written under the form:

$$J(\Omega_{x,\varepsilon}) = J(\Omega) + \varepsilon^d s_\Omega(x) g_\Omega(x) + o(\varepsilon^d), \text{ where } s_\Omega(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ -1 & \text{if } x \in D \setminus \overline{\Omega}. \end{cases}$$

- The first-order necessary conditions for optimality of a shape Ω then read:

$$\begin{cases} \forall x \in \Omega, & g_\Omega(x) > 0, \\ \forall x \in D \setminus \overline{\Omega}, & g_\Omega(x) < 0, \end{cases}$$

that is:

$$-g_\Omega \text{ is one Level Set function for } \Omega.$$

- A fixed point algorithm with relaxation is implemented to enforce this condition:

Search for a Level Set function $\phi : D \to \mathbb{R}$ s.t. $||\phi||_{L^2(D)} = 1$ and

$$\phi = -\frac{1}{||g_\Omega||_{L^2(D)}} g_\Omega, \text{ where } \Omega := \{x \in D,\ \phi(x) < 0\}.$$

S. Amstutz' algorithm [AmHan, Am] (II)

Initialization: Initial Level Set function ϕ^0,

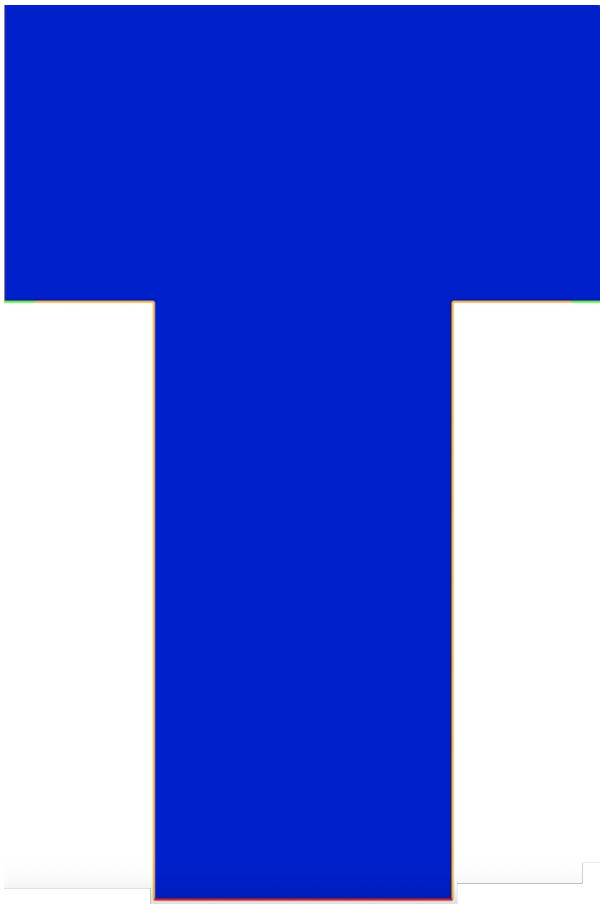
For $n = 0, \ldots$ **until convergence:**

1. Calculate the state u_{Ω^n} and adjoint p_{Ω^n};
2. Calculate $g_{\Omega^n}(x)$ and $g^n := \frac{1}{||g_{\Omega^n}||_{L^2(D)}} g_{\Omega^n}$.
3. Calculate $a^n := \arccos(\phi^n, g^n)_{L^2(D)}$;
4. Choose a relaxation step $0 < \tau^n < 1$;
5. The updated Level Set function ϕ^{n+1} is obtained by spherical interpolation:

$$\phi^{n+1} = \frac{\sin((1-\tau^n)a^n)\phi^n + \sin(\tau^n a^n)g^n}{\sin a^n},$$

corresponding to the new shape

$$\Omega^{n+1} = \{\phi^{n+1} < 0\}.$$



An algorithm combining topological derivatives with a diffusion equation

- The continuous-in-time shape $\Omega(t)$ is represented by a Level Set function $\phi(t, x)$.
- The shape evolves through a diffusion equation:

$$\frac{\partial \phi}{\partial t} - \underbrace{\varepsilon^2 \kappa(\phi) \Delta \phi}_{\text{Diffusion term}} = \kappa(\phi) \underbrace{\mathrm{d}_T J(\Omega)}_{\text{Source term}} .$$

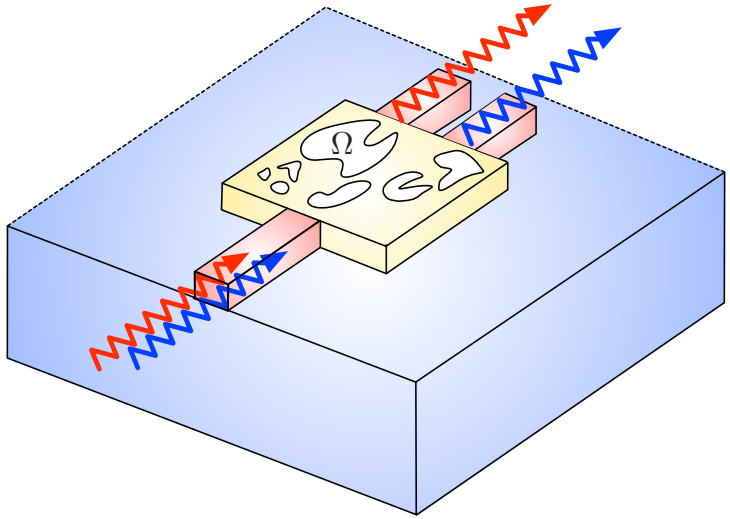
- Like the perimeter in phase-field methods, the diffusion term smoothes the Level Set function over a length scale ε.
- The topological derivative as source term imposes that:
  - ϕ decreases where $\mathrm{d}_T J(\Omega)(x) < 0$;
  - ϕ increase where $\mathrm{d}_T J(\Omega)(x) > 0$.
- The coefficient $\kappa(\phi)$ may be tuned depending on the situation.

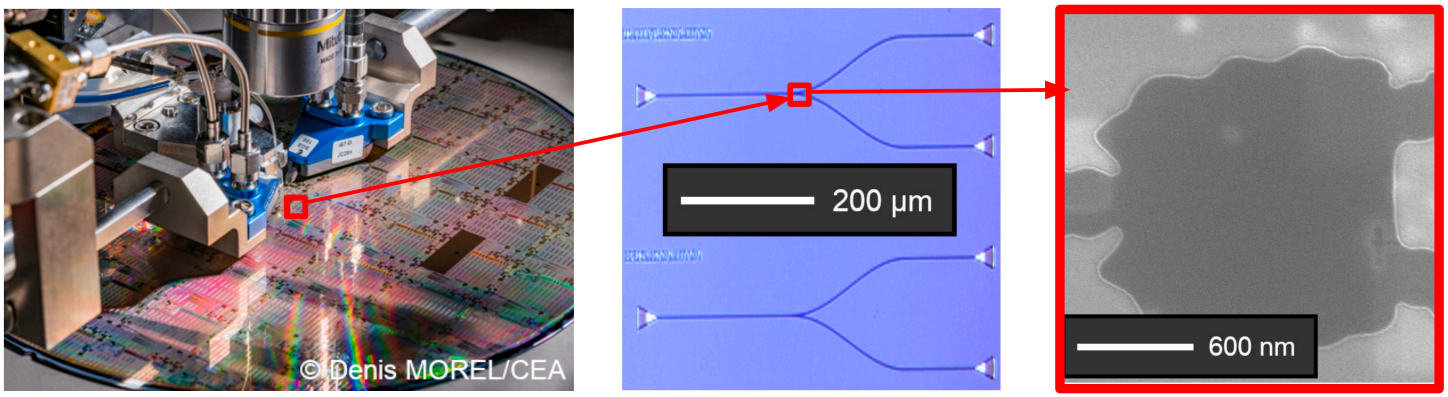
T. Yamada, S. Nishiwaki & A. Takezawa, *A topology optimization method based on the level set method incorporating a fictitious interface energy*, Computer Methods in Applied Mechanics and Engineering, 199 (45-48), (2010), pp. 2876–2891.

Part II

Some applications of the Level Set Method

- Nanophotonic devices are the basic components of photonic integrated circuits.
- In these, light is transported by wave guides.
- The electric and magnetic fields are governed by the time-harmonic Maxwell's equations.
- The shape Ω of air inclusions in the Si core is optimized to achieve particular effects.

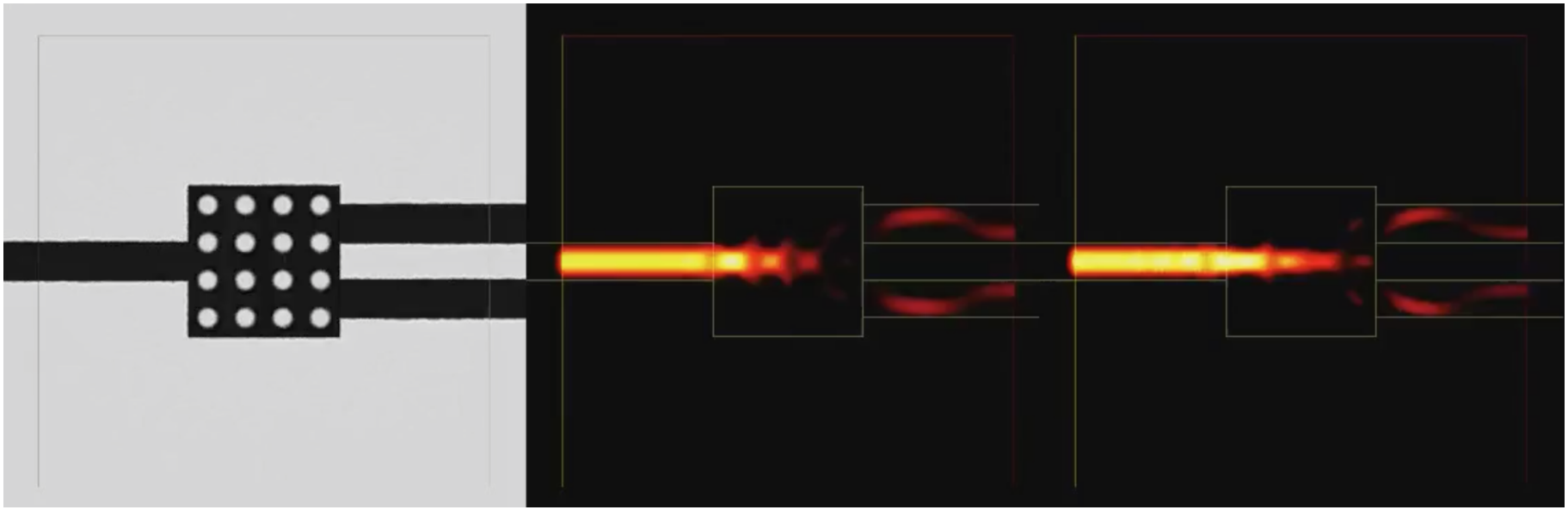




One nanophotonic component inside a complete photonic circuit.

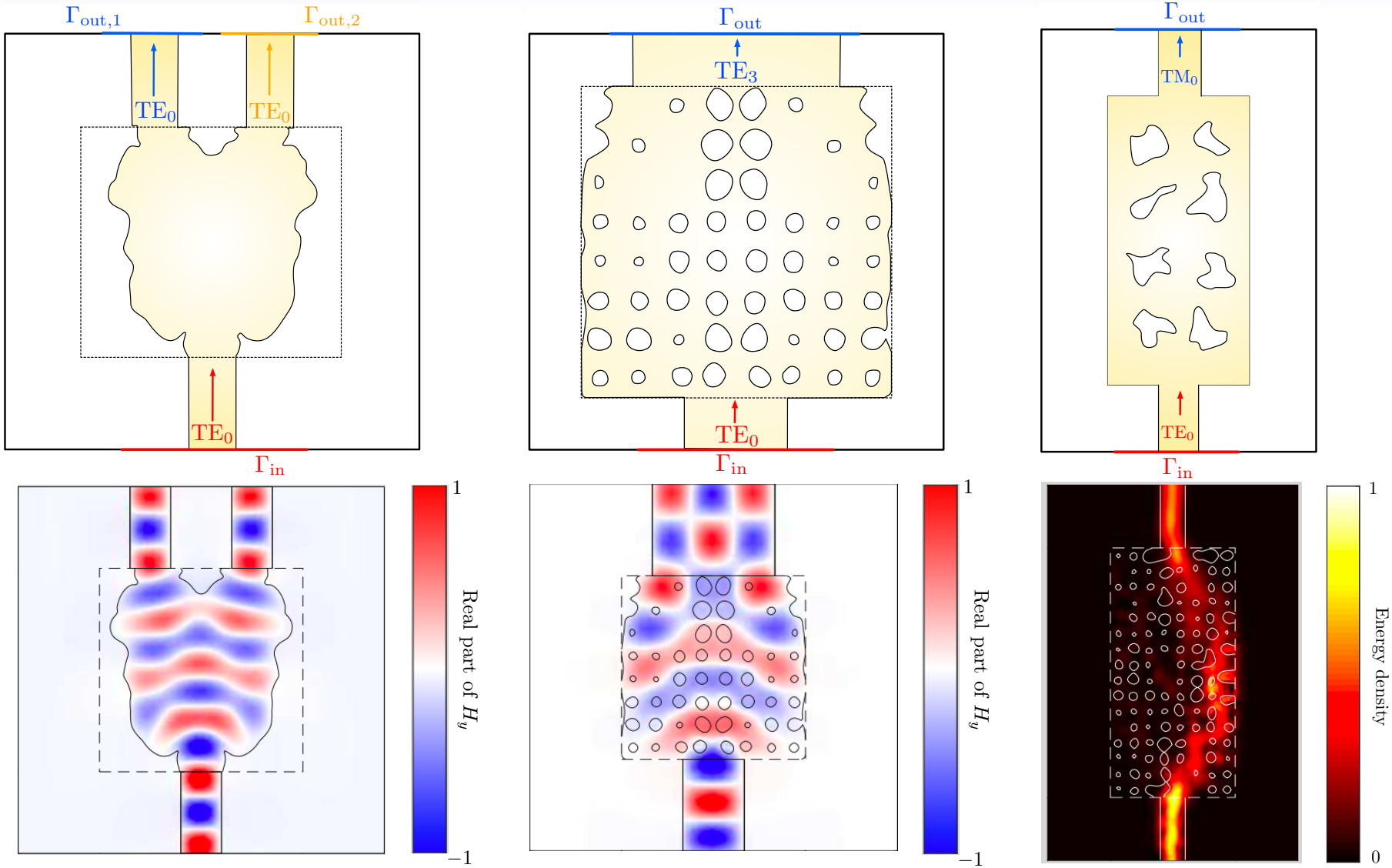
Optimization of nanophotonic devices (II)

Duplexers steer incoming waves to different output channels, depending on their wavelength.



Optimization of the shape of a nanophotonic duplexer.

Optimization of other nanophotonic devices

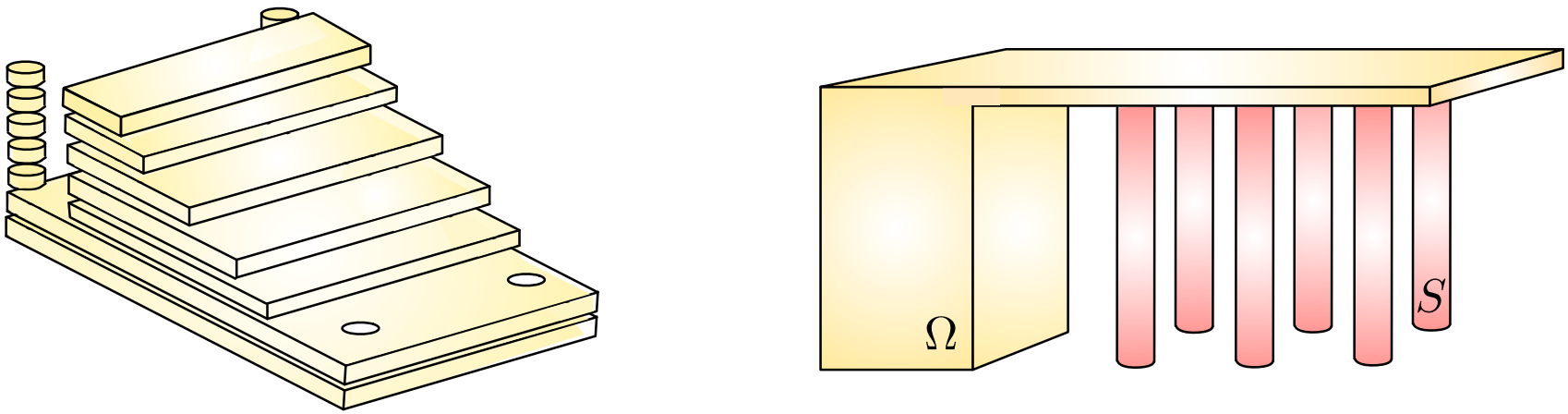


Optimization of (left) A power divider; (center) A mode converter; (right) A polarization converter.

Part II

Some applications of the Level Set Method

- Additive manufacturing techniques assemble shapes in a layer-by-layer fashion.
- Although they allow to proceed complex shapes, they usually experience difficulties with large overhangs, i.e. horizontal regions hanging over void.
- To assemble such patterns, a support structure S is often erected alongside Ω.
- This structure has to be removed with a machine tool before the final use of Ω.



(Left) Layer-by-layer assembly of a shape; (Right) Support structure for the assembly of an overhang region.

Structural optimization under accessibility constraints (II)

- We optimize the shapes Ω of a 3d bridge and S of its support structure:

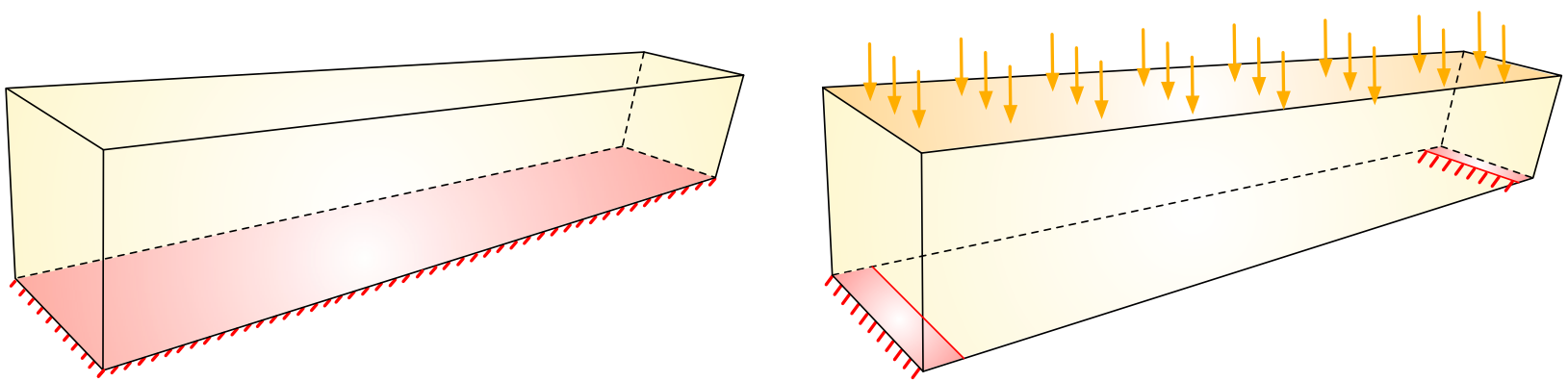
$$\min_{\Omega,S} J(\Omega,S), \text{ s.t. } \begin{cases} \mathrm{Vol}(\Omega) \leq V_{\Omega,\mathsf{T}}, \\ \mathrm{Vol}(S) \leq V_{S,\mathsf{T}}. \end{cases}$$

- The objective function reads:

$$J(\Omega,S) = C(\Omega) + C_{\mathsf{man}}(\Omega,S),$$

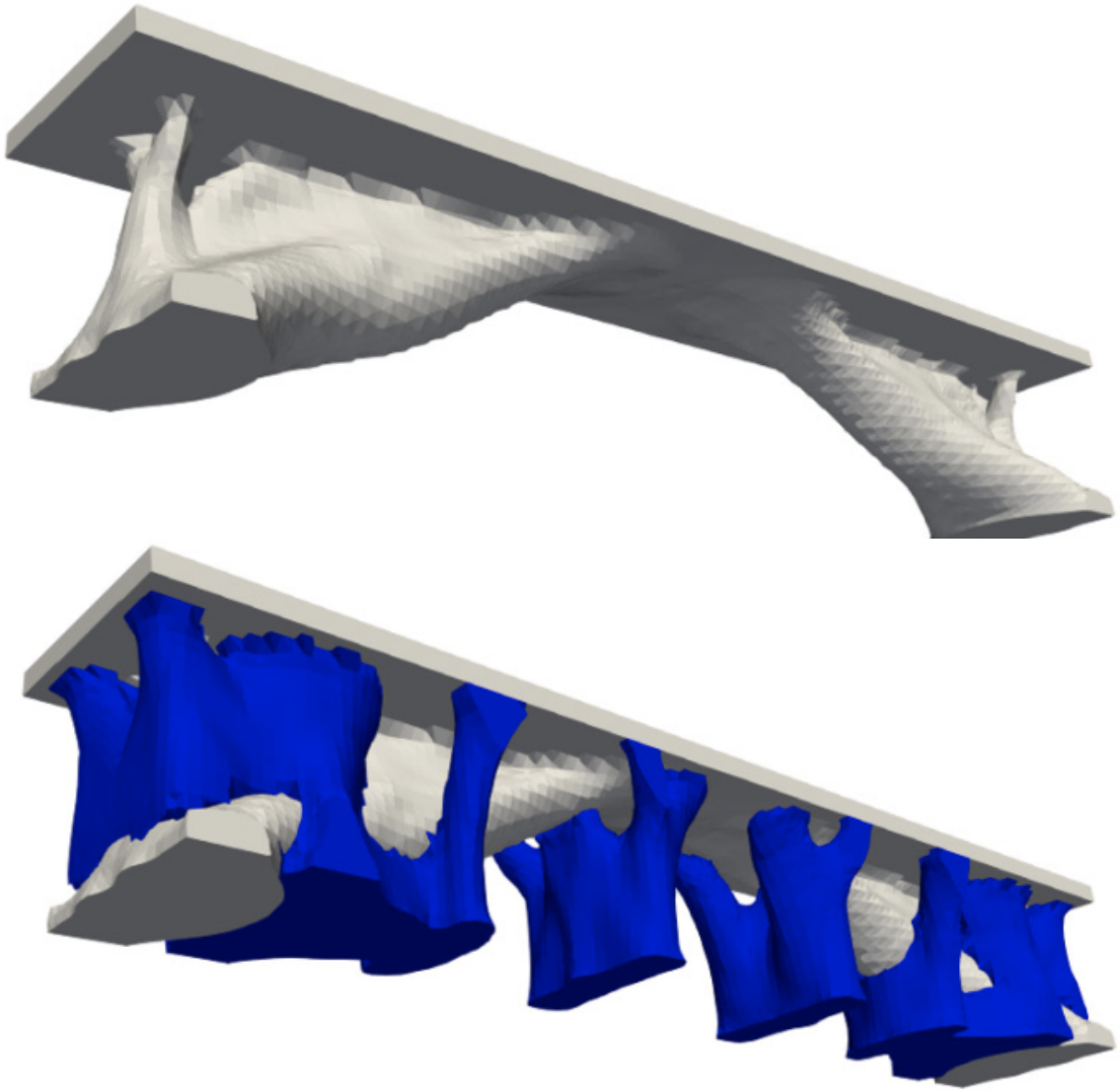
where:

- $C(\Omega)$ is the compliance of Ω in its use-case;
- $C_{\mathsf{man}}(\Omega,S)$ is the compliance of the total structure $\Omega \cup S$ during construction.



Physical boundary conditions on (left) The support structure; (right) The bridge in its use situation.

Structural optimization under accessibility constraints (III)

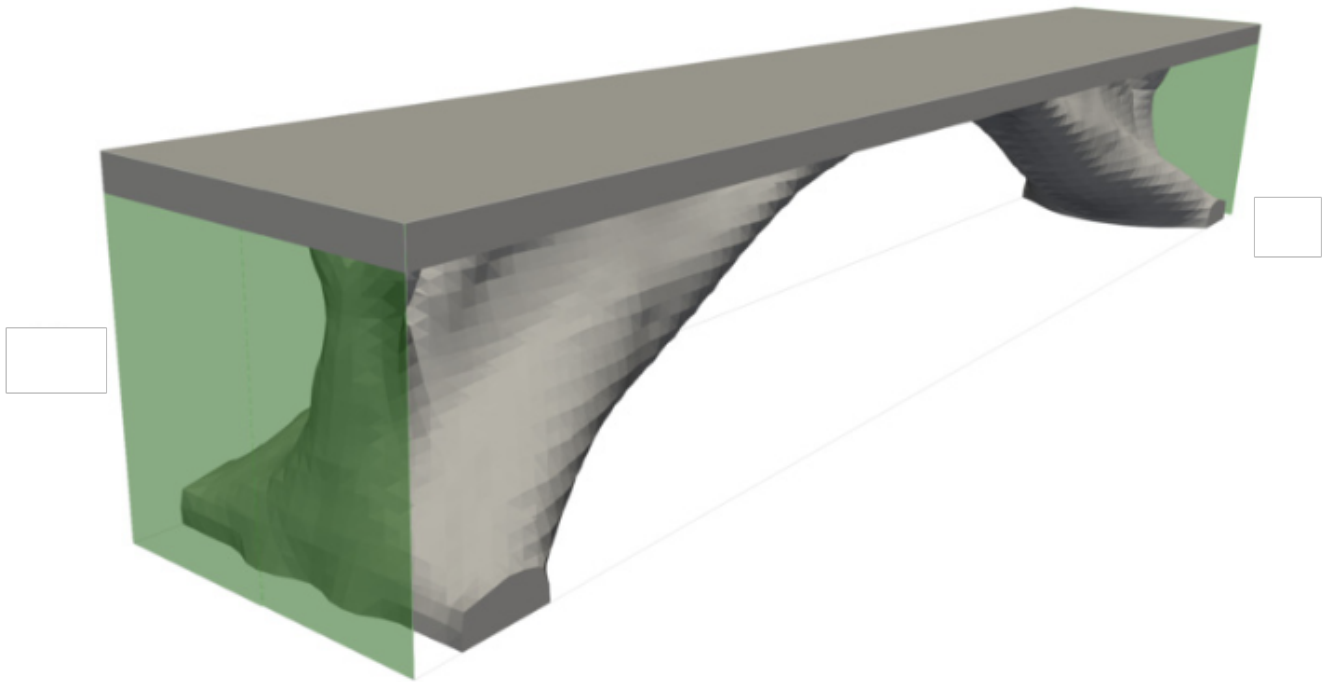


Optimized shape of the bridge Ω (in grey) and its supporting structure S (in blue) with respect to their end-use and manufacturing compliances.

Structural optimization under accessibility constraints (IV)

Problem:

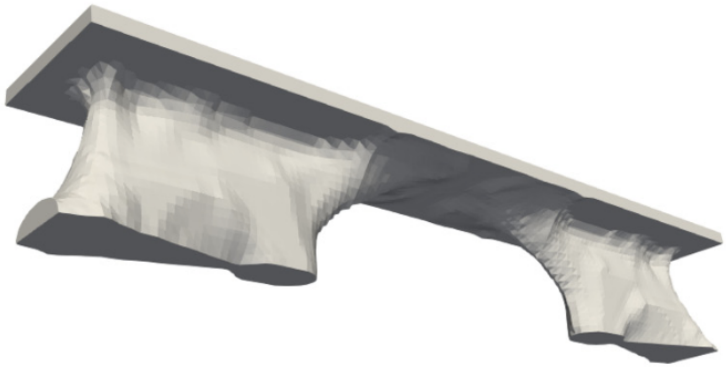
- After construction, the support structure S has to be removed by a machine tool;
- So that it is possible, it must be accessible from a given entry region.

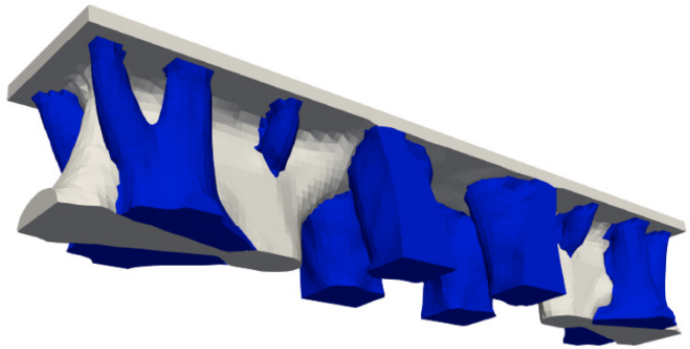


Supports have to be accessible from one of the transparent green sides

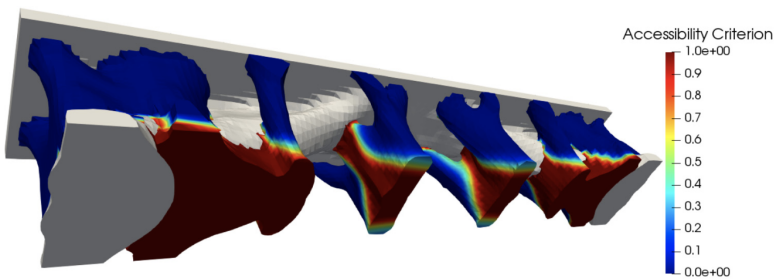
- We add an accessibility constraint on S from one of the green sides.
- The latter is modeled from the signed distance function d_Ω.

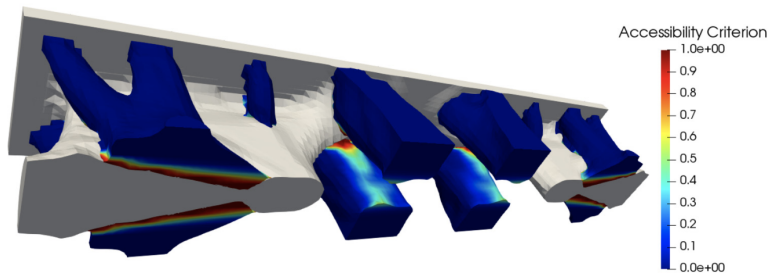
Structural optimization under accessibility constraints (V)





(Left) Optimized shape of the bridge Ω (in grey) and its supporting structure S (in blue), taking into account accessibility constraints.





Values of the accessibility functional (Left) Without, (right) While taking into account accessibility constraints.

The Level Set shape and topology optimization framework: conclusions

Assets of the method:

- The Level Set Method is efficient and compatible with industrial requirements:
  - It is cheap: the total CPU cost is essentially that of the PDE solver;
  - It can be used on arbitrary design domain meshes;
  - It is readily combined with constrained optimization solvers.
- The variable is the shape $\Omega \subset \mathbb{R}^d$. This allows to formulate naturally "difficult" geometric constraints about, e.g.
  - The thickness and castability of Ω [AlJouMi],
  - The absence of overhang features [AlDaEs],
  - Its accessibility to manufacturing tools [AlBiBoGo].

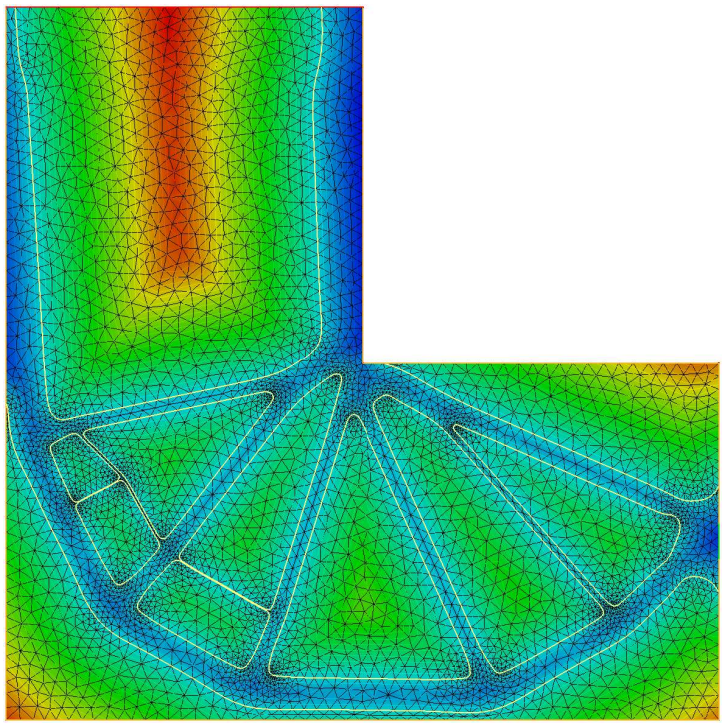
Drawback:

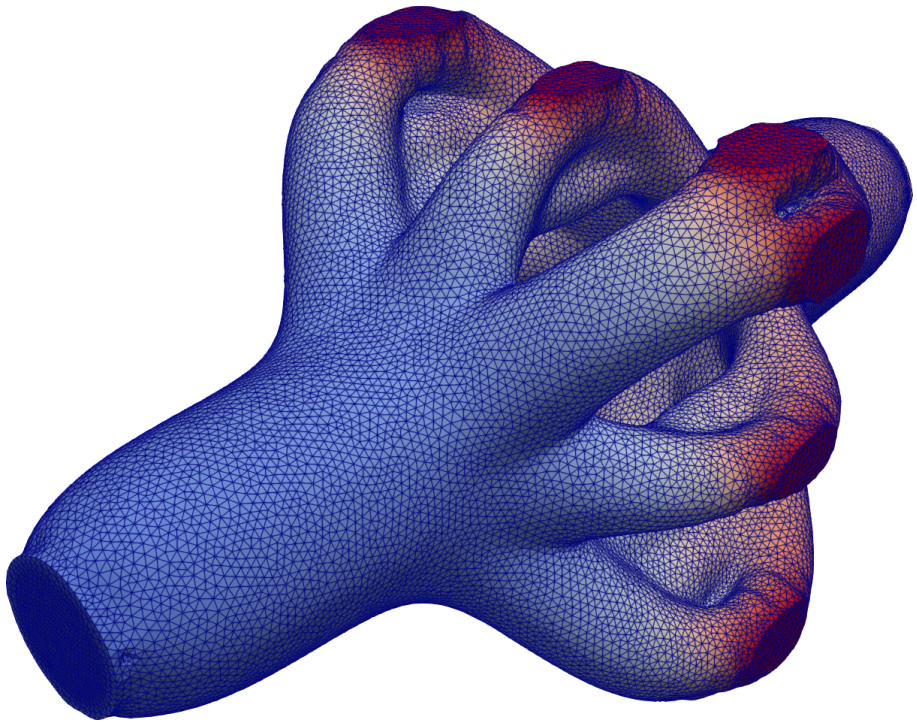
- Like any fixed mesh method, the solution of physical PDEs on Ω is not straightforward. This task can be addressed by:
  - Approximations of the PDE, e.g. by the ersatz method in elasticity.
  - Use of intrusive numerical solvers: `X-FEM` or `CutFEM` [ViMau, ViMau2].
- The shape gradient is calculated at the continuous level, then discretized: the convergence of the method is never "perfect".

Program of the next webinar

`Webinar 3` A body-fitted variant of the Level Set Method.

- A taste of remeshing;
- Presentation of the method;
- Application examples.





Thank you for your attention!

Technical appendix

The Sobolev space $H^1(\Omega)$

Let $\Omega \subset \mathbb{R}^d$ be open, and let $L^2(\Omega)$ be the space of square integrable functions on Ω.

- The Sobolev space $H^1(\Omega)$ is defined by:

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \text{ s.t. } u \text{ has a weak derivative } \frac{\partial u}{\partial x_i} \in L^2(\Omega) \text{ for } i = 1, \dots, d \right\}.$$

- This space is a Hilbert space for the inner product and norms:

$$\langle u, v \rangle_{H^1(\Omega)} = \int_\Omega uv \, \mathrm{d}x + \sum_{i=1}^{d} \int_\Omega \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, \mathrm{d}x, \text{ and } ||u||_{H^1(\Omega)} := \langle u, u \rangle_{H^1(\Omega)}^{1/2}.$$

Reminder:

- Let $u \in L^2(\Omega)$ and $i \in \{1, \dots, d\}$; if there exists a function $w_i \in L^2(\Omega)$ such that:

$$\forall \varphi \in \mathcal{C}_c^\infty(\Omega), \quad \int_\Omega u \frac{\partial \varphi}{\partial x_i} \, \mathrm{d}x = - \int_\Omega w_i \varphi \, \mathrm{d}x.$$

  then u has weak (or distributional) partial derivative $w_i := \frac{\partial u}{\partial x_i} \in L^2(\Omega)$.

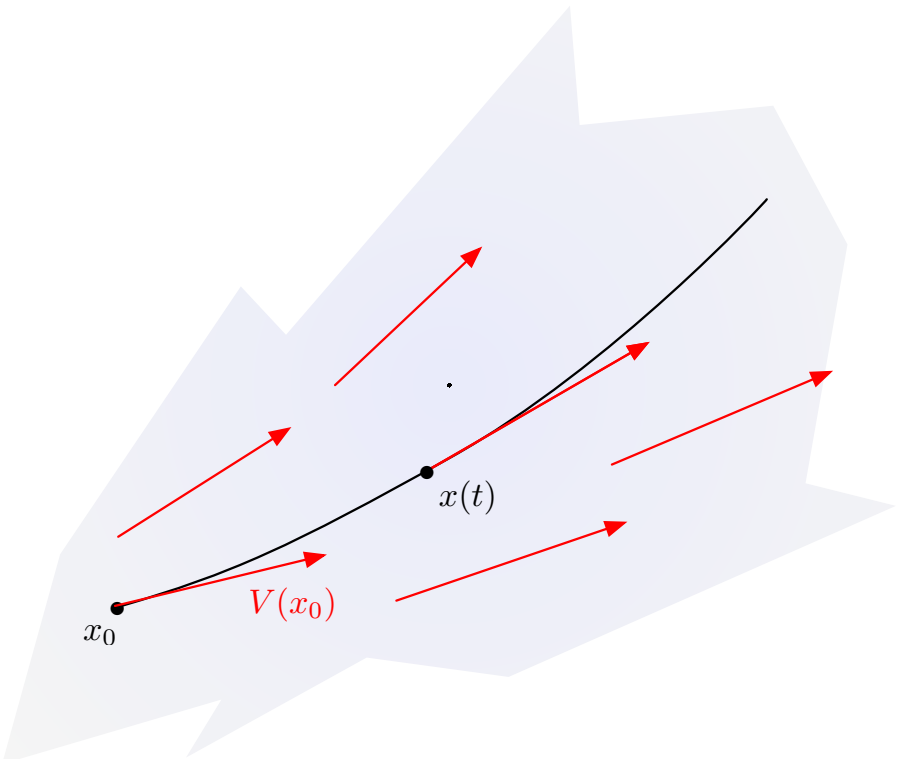
- If $u \in \mathcal{C}_c^1(\Omega)$, then u has weak derivatives which equal its classical derivatives.

Runge-Kutta integration of dynamical systems (I)

Let $V : \mathbb{R}^d \to \mathbb{R}^d$ be a (smooth) vector field; we consider the dynamical system

$$\begin{cases} x'(t) = V(x(t)) & \text{for } t \in (0, T), \\ x(0) = x_0, \end{cases}$$

for the trajectory $t \mapsto x(t)$ of a particle with velocity V.

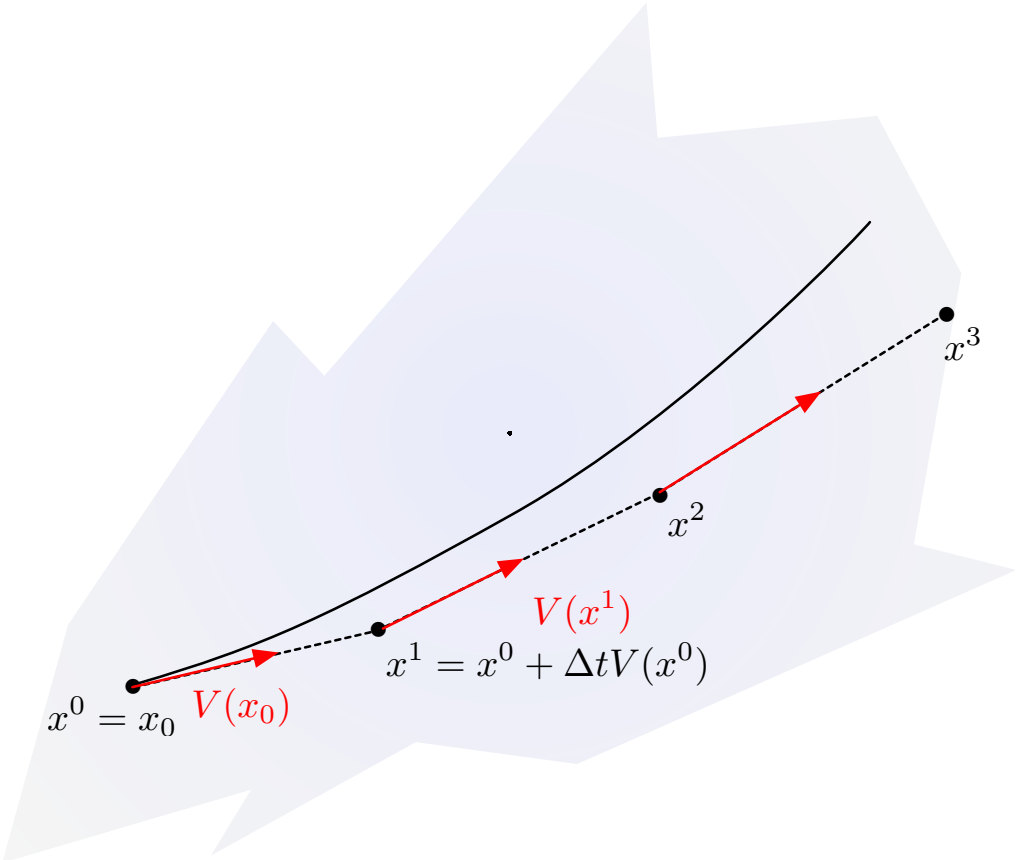


Introducing a subdivision $t^n = n\Delta t$ of $(0, T)$, $n = 0, \ldots, N := T/\Delta t$, we aim to calculate an approximation x^n of $x(t^n)$.

Runge-Kutta integration of dynamical systems (II)

The first-order, explicit Euler approximation of this dynamical system reads:

$$\begin{cases} x^{n+1} = x^n + \Delta t V(x^n) & \text{for } n = 0, \ldots, N-1, \\ x^0 = x_0. \end{cases}$$



This method is only first-order accurate as $\Delta t \to 0$:

$$\forall n \in 0, \ldots, N, \quad |x(t^n) - x^n| \leq C \Delta t \text{ for some constant } C > 0.$$

Runge-Kutta integration of dynamical systems (III)

According to the Runge-Kutta 2 method, the iterate x^{n+1} is obtained from x^n by:

1. An attempt step is performed with the 1st-order Euler method:

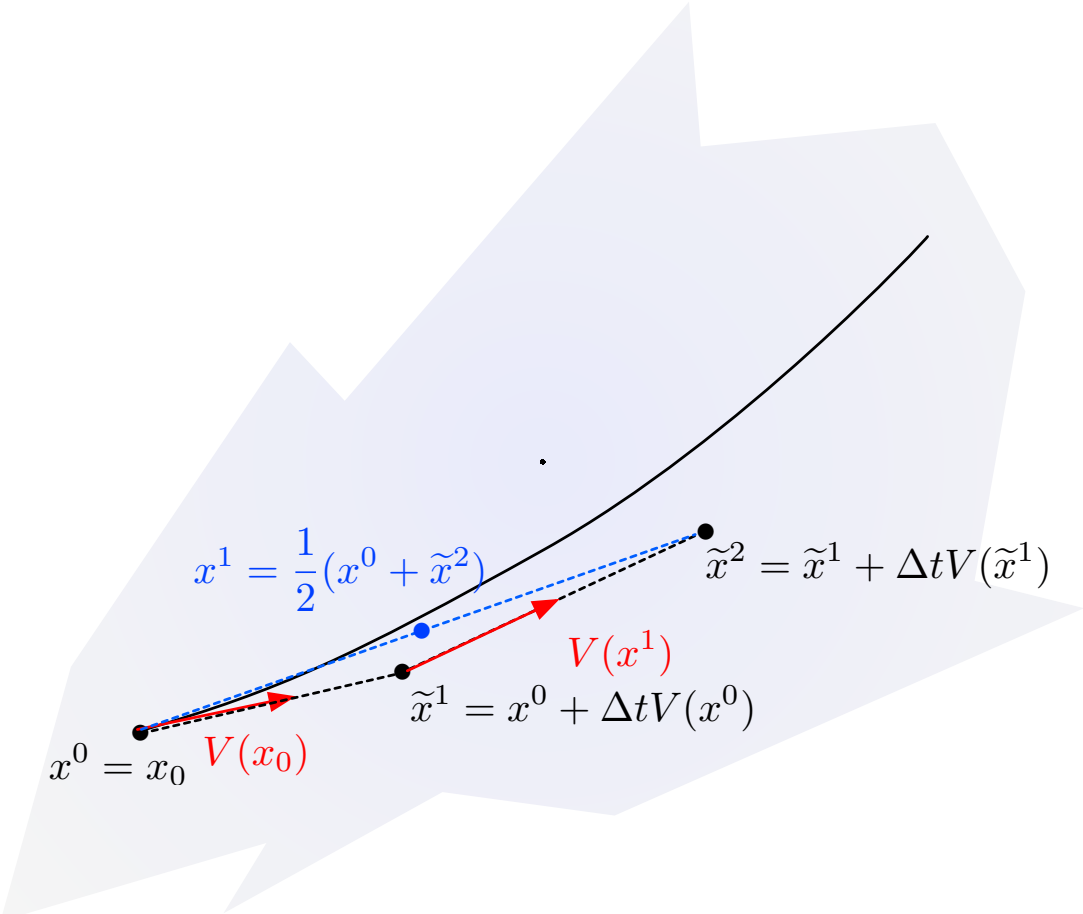
$$\widetilde{x}^{n+1} := x^n + \Delta t V(x^n).$$

2. Another attempt step is performed from $\widetilde{x}^{n+1}$:

$$\widetilde{x}^{n+2} := \widetilde{x}^{n+1} + \Delta t V(\widetilde{x}^{n+1}).$$

3. The point x^{n+1} is obtained by averaging:

$$x^{n+1} = \frac{1}{2}(x^n + \widetilde{x}^{n+2}).$$



This method is second-order accurate:

$$\forall n = 0, \ldots, N, \quad |x(t^n) - x^n| \leq C\Delta t^2 \text{ for some constant } C > 0.$$

Bibliography

References I

[Abgrall] R. Abgrall, *Numerical Discretization of the First-Order Hamilton-Jacobi Equation on Triangular Meshes*, Com. Pure Applied Math. XLIX, (1996) pp. 1339–1373.

[AlBiBoGo] G. Allaire, M. Bihr, B. Bogosel and M. Godoy, *Accessibility constraints in structural optimization via distance functions*, Journal of Computational Physics, 484, (2023), 112083.

[AlDeJouToa] G. Allaire, F. de Gournay, F. Jouve and A.-M. Toader, *Structural optimization using topological and shape sensitivity via a level set method*, Control and cybernetics, 34 (1), pp. 59–80.

[AlDaJou] G. Allaire, C. Dapogny, and F. Jouve, *Shape and topology optimization*, in Geometric partial differential equations, part II, A. Bonito and R. Nochetto eds., Handbook of Numerical Analysis, vol. 22, (2021), pp.1-132.

[AlDaEs] G. Allaire, C. Dapogny, R. Estevez, A. Faure and G. Michailidis, *Structural optimization under overhang constraints imposed by additive manufacturing technologies*, Journal of Computational Physics, 351, (2017), pp. 295–328.

References II

[AlJouMi] G. Allaire, F. Jouve, and G. Michailidis, *Thickness control in structural optimization via a level set method*, Structural and Multidisciplinary Optimization, 53(6), (2016), pp. 1349–1382.

[AmHan] S. Amstutz and H. Andra, *A new algorithm for topology optimization using a level-set method*, Journal of Computational Physics, 216 (2), (2006), pp. 573–588.

[Am] S. Amstutz, *Analysis of a level set method for topology optimization*, Optimization Methods and Software, 26(4-5), (2011), pp. 555–573.

[AWu] H. Azegami and Z. C. Wu, *Domain optimization analysis in linear elastic problems: approach using traction method*, JSME international journal. Ser. A, Mechanics and material engineering, 39(2), (1996), pp. 272–278.

[BaSe] T.J. Barth and J.A. Sethian, *Numerical Schemes for the Hamilton–Jacobi and Level Set Equations on Triangulated Domains*, J. Comput. Phys., 145 (1998) pp. 1–40.

References III

[BlanGar] L. Blank, H. Garcke, L. Sarbu, T. Srisupattarawanit, V. Styles and A. Voigt, *Phase-field approaches to structural topology optimization*, In Constrained optimization and optimal control for partial differential equations, (2011), pp. 245–256.

[BouCha] B. Bourdin and A. Chambolle, *Design-dependent loads in topology optimization*, ESAIM: Control, Optimisation and Calculus of Variations, 9 (2003), pp. 19–48.

[BuDaFre] C. Bui, C. Dapogny and P. Frey, *An accurate anisotropic adaptation method for solving the level set advection equation*, International Journal for Numerical Methods in Fluids, 70(7), (2012), pp. 899–922.

[Bu] M. Burger, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.

[Ce] J. Céa, *Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût*, ESAIM: Modélisation mathématique et analyse numérique, 20(3), (1986), pp. 371–402.

References IV

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

[Err] R. M. Errico, *What is an adjoint model?*, Bulletin of the American Meteorological Society, 78(11), (1997), pp. 2577–2592.

[FeAlDa] F. Feppon, G. Allaire and C. Dapogny, *Null space gradient flows for constrained optimization with applications to shape optimization*, ESAIM: Control, Optimization and Calculus of Variations, 26, (2020).

[GiaPanTra] M. Giacomini, O. Pantz and K. Trabelsi, *Volumetric expressions of the shape gradient of the compliance in structural shape optimization*, `arXiv` preprint `arXiv:1701.05762` (2017).

[GiPier] M. B. Giles and N. A. Pierce, *An introduction to the adjoint approach to design*, Flow, turbulence and combustion, 65(3), (2000), pp .393–415.

References V

[Gou] F. de Gournay, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM journal on control and optimization, 45 (2006), pp. 343–367.

[KiSe] R. Kimmel and J.A. Sethian, *Computing geodesic paths on manifolds*, Proceedings of the national academy of Sciences, 95(15), (1998), pp.8431–8435.

[LeDa] N. Lebbe, C. Dapogny, E. Oudet, K. Hassan, & A. Gliere, *Robust shape and topology optimization of nanophotonic devices using the level set method*, Journal of Computational Physics, 395, (2019), pp. 710–746.

[Li] J.-L. Lions, *Optimal control of systems governed by partial differential equations*, Springer, (1971).

[MoPir] B. Mohammadi et O. Pironneau, *Applied shape optimization for fluids*, 2nd edition, Oxford University Press, (2010).

[Neu] J. Neuberger, *Sobolev gradients and differential equations*, Springer Science & Business Media, (2009).

References VI

[NoWri] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Science, (1999).

[Pironneau] O. Pironneau, *Finite element methods for fluids*, Chichester: Wiley, (1989).

[Ple] R. E. Plessix, *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*, Geophysical Journal International, 167(2), (2006), pp. 495–503.

[Strain] J. Strain, *Semi-Lagrangian methods for level set equations*, Journal of Computational Physics, 151(2), (1999), pp. 498–533.

[ViMau] C. H. Villanueva and K. Maute, *Density and level set-XFEM schemes for topology optimization of 3-D structures*, Computational Mechanics, 54(1), (2014), pp. 133–150.

[ViMau2] C. H. Villanueva and K. Maute, *CutFEM topology optimization of 3D laminar incompressible flow problems*, Computer Methods in Applied Mechanics and Engineering, 320, (2017), pp. 444–473.

References VII

[YaNiTa] T. Yamada, S. Nishiwaki & A. Takezawa, *A topology optimization method based on the level set method incorporating a fictitious interface energy*, Computer Methods in Applied Mechanics and Engineering, 199 (45-48), (2010), pp. 2876–2891.