Null Space Gradient Flows for Constrained Optimization with Applications to Shape Optimization

Florian Feppon

Grégoire Allaire, Charles Dapogny
Julien Cortial, Felipe Bordeu

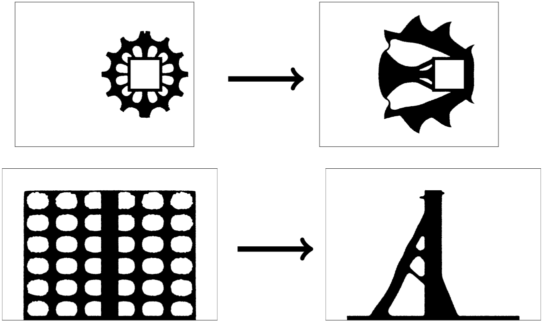
SIAM CSE – Spokane – February 26, 2019

SAFRAN

l'X ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

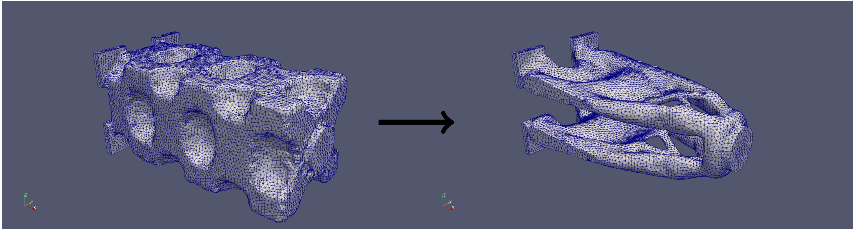
Shape optimization

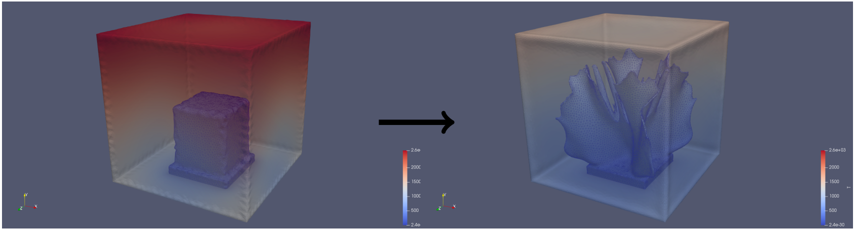
Multiphysics, non parametric, shape and topology optimization, in 2D. . .



Shape optimization

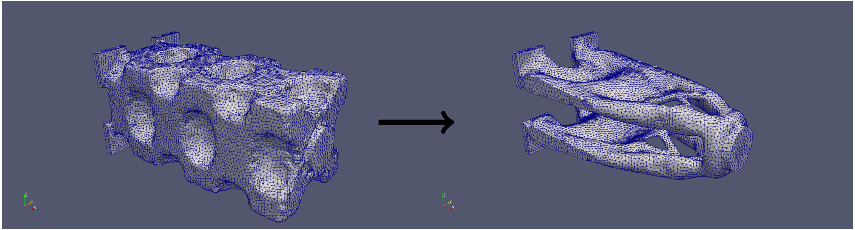
And in 3D...

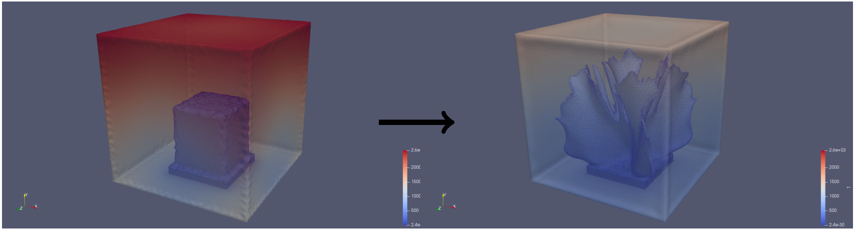




Shape optimization

And in 3D. . .





. . . Nonlinear, non convex, infinite dimensional optimization problems featuring multiple and arbitrary constraints!

Outline

1. Gradient flows for equality and *inequality* constrained optimizations
2. Demonstration on topology optimization test cases

1. Constrained optimization

$$\min_{x \in \mathcal{X}} \quad J(x)$$
$$\text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases}$$

with $J : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^q$ Fréchet differentiable.
The set $\mathcal{X}$ can be

- a finite dimensional vector space, $\mathcal{X} = \mathbb{R}^n$

1. Constrained optimization

$$\min_{x \in \mathcal{X}} \quad J(x)$$
$$\text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases}$$

with $J : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^q$ Fréchet differentiable.
The set $\mathcal{X}$ can be

- a finite dimensional vector space, $\mathcal{X} = \mathbb{R}^n$
- a Hilbert space equipped with a scalar product $a(\cdot, \cdot)$, $\mathcal{X} = V$

1. Constrained optimization

$$\min_{x\in\mathcal{X}} \quad J(x)$$
$$\text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases}$$

with $J : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^q$ Fréchet differentiable.
The set $\mathcal{X}$ can be

- a finite dimensional vector space, $\mathcal{X} = \mathbb{R}^n$
- a Hilbert space equipped with a scalar product $a(\cdot,\cdot)$, $\mathcal{X} = V$
- a "manifold", as in topology optimization:

$$\mathcal{X} = \{\Omega \subset D \,|\, \Omega \text{ Lipschitz }\}$$

1. A generic optimization algorithm

From a current guess x_n, how to select the descent direction $\boldsymbol{\xi}_n$ given objective J and constraints $\boldsymbol{g}$, $\boldsymbol{h}$?

1. A generic optimization algorithm

From a current guess x_n, how to select the descent direction $\boldsymbol{\xi}_n$ given objective J and constraints $\boldsymbol{g}$, $\boldsymbol{h}$?

- Many "iteratives" methods in literature:
  - Penalty methods (like Augmented Lagrangian Method)
  - Linearization methods : SLP, SQP, MMA, MFD

1. A generic optimization algorithm

From a current guess x_n, how to select the descent direction $\boldsymbol{\xi}_n$ given objective J and constraints $\boldsymbol{g}$, $\boldsymbol{h}$?

- Many "iteratives" methods in literature:
  - Penalty methods (like Augmented Lagrangian Method)
  - Linearization methods : SLP, SQP, MMA, MFD

These methods suffer from:

- the need for tuning unintuitive parameters.

1. A generic optimization algorithm

From a current guess x_n, how to select the descent direction $\boldsymbol{\xi}_n$ given objective J and constraints $\boldsymbol{g}$, $\boldsymbol{h}$?

- Many "iteratives" methods in literature:
  - Penalty methods (like Augmented Lagrangian Method)
  - Linearization methods : SLP, SQP, MMA, MFD

These methods suffer from:

- the need for tuning unintuitive parameters.
- "inconsistencies" when $\Delta t \to 0$: SLP, SQP, MFD subproblems may not have a solution if Δt too small. ALG does not guarantee reducing the objective function if constraints are satisfied.

1. A generic optimization algorithm

Dynamical systems approaches:

1. A generic optimization algorithm

Dynamical systems approaches:

- For unconstrained optimization, the celebrated gradient flow:

$$\dot{x} = -\nabla J(x)$$

  $J(x(t))$ decreases necessarily!

1. A generic optimization algorithm

Dynamical systems approaches:

- For unconstrained optimization, the celebrated gradient flow:

  $$\dot{x} = -\nabla J(x)$$

  $J(x(t))$ decreases necessarily!

- For equality constrained optimization, projected gradient flow (Tanabe (1980)):

  $$\dot{x} = -(I - \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x)$$

  (gradient flow on $\mathcal{X} = \{x \in V \mid \boldsymbol{g}(x) = 0\}$)

1. A generic optimization algorithm

Dynamical systems approaches:

- For unconstrained optimization, the celebrated gradient flow:

  $$\dot{x} = -\nabla J(x)$$

  $J(x(t))$ decreases necessarily!
- For equality constrained optimization, projected gradient flow (Tanabe (1980)):

  $$\dot{x} = -(I - \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x)$$

  (gradient flow on $\mathcal{X} = \{x \in V \,|\, \boldsymbol{g}(x) = 0\}$)
  Then Yamashita (1980) added a Gauss-Newton direction:

  $$\dot{x} = -\alpha_J(I - \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x) - \alpha_C \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\boldsymbol{g}(x)$$

1. A generic optimization algorithm

Dynamical systems approaches:

- For unconstrained optimization, the celebrated gradient flow:

$$\dot{x} = -\nabla J(x)$$

  $J(x(t))$ decreases necessarily!
- For equality constrained optimization, projected gradient flow (Tanabe (1980)):

$$\dot{x} = -(I - \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x)$$

  (gradient flow on $\mathcal{X} = \{x \in V \,|\, \boldsymbol{g}(x) = 0\}$)
  Then Yamashita (1980) added a Gauss-Newton direction:

$$\dot{x} = -\alpha_J(I - \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x) - \alpha_C\mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\boldsymbol{g}(x)$$

  $\boldsymbol{g}(x(t)) = \boldsymbol{g}(x(0))e^{-\alpha_C t}$ and $J(x(t))$ is guaranteed to decrease if $\boldsymbol{g}(x(t)) = 0$.

1. A generic optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, consider $\widetilde{I}(x)$ the set of violated constraints:

$$\widetilde{I}(x) = \{i \in \{1, \ldots, q\} \mid h_i(x) \geqslant 0\}.$$

$$\boldsymbol{C}_{\widetilde{I}(x)} = \left[\boldsymbol{g}(x) \quad | \quad (h_i(x))_{i \in \widetilde{I}(x)}\right]^T$$

1. A generic optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, consider $\widetilde{I}(x)$ the set of violated constraints:

$$\widetilde{I}(x) = \{i \in \{1, \ldots, q\} \mid h_i(x) \geqslant 0\}.$$

$$\boldsymbol{C}_{\widetilde{I}(x)} = \left[\boldsymbol{g}(x) \quad | \quad (h_i(x))_{i \in \widetilde{I}(x)}\right]^T$$

We propose

$$\dot{x} = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t))$$

with

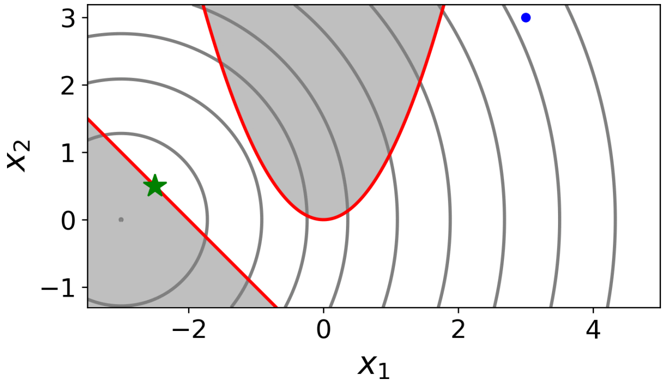
$$-\boldsymbol{\xi}_J(x) := \begin{cases} \text{the "best" descent direction} \\ \text{with respect to the constraints } \widetilde{I}(x) \end{cases}$$

$$-\boldsymbol{\xi}_C(x) := \begin{cases} \text{the Gauss-Newton direction} \\ -\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^{\mathcal{T}})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x) \end{cases}$$

1. A generic optimization algorithm

$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad J(x_1,x_2) = x_1^2 + (x_2+3)^2$$

$$\text{s.t.} \quad \begin{cases} h_1(x_1,x_2) = -x_1^2 + x_2 & \leq 0 \\ h_2(x_1,x_2) = -x_1 - x_2 - 2 & \leq 0 \end{cases}$$



1. A generic optimization algorithm

We propose:

$$\dot{x} = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t))$$

with

$$\boldsymbol{\xi}_J(x) := (I - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})(\nabla J(x))$$

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^{\mathcal{T}})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x).$$

1. A generic optimization algorithm

We propose:

$$\dot{x} = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t))$$

with

$$\boldsymbol{\xi}_J(x) := (I - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})(\nabla J(x))$$

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x).$$

$\widehat{I}(x) \subset \widetilde{I}(x)$ is a subset of the active or violated constraints which can be computed by mean of a dual subproblem.

1. A generic optimization algorithm

The best descent direction $-\boldsymbol{\xi}_J(x)$ must be proportional to

$$\boldsymbol{\xi}^* = \quad \arg\min_{\boldsymbol{\xi} \in V} \quad \mathrm{D}J(x)\boldsymbol{\xi}$$

$$\text{s.t.} \begin{cases} \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0 \\ \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \leq 0 \\ ||\boldsymbol{\xi}||_V \leq 1. \end{cases}$$

where $\boldsymbol{h}_{\widetilde{I}(x)}(x) = (h_i(x))_{i \in \widetilde{I}(x)}$

Proposition

Let $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) \in \mathbb{R}^p \times \mathbb{R}^{\mathrm{Card}\widetilde{I}(x)}$ *the solutions of the following dual minimization problem:*

$$(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) := \arg \min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^p \\ \boldsymbol{\mu}\in\mathbb{R}^{\widetilde{q}(x)},\, \boldsymbol{\mu}\geqslant 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}||_V.$$

Then, unless x *is a KKT point, the best descent direction* $\boldsymbol{\xi}^*(x)$ *is given by*

$$\boldsymbol{\xi}^*(x) = -\frac{\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}^*(x)}{||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}^*(x)||_V}.$$

Proposition

Let $\widehat{I}(x)$ the set obtained by collecting the non zero components of $\boldsymbol{\mu}^*(x)$:

$$\widehat{I}(x) := \{i \in \widetilde{I} \,|\, \mu_i^*(x) > 0\}.$$

Then $\boldsymbol{\xi}^*(x)$ is explicitly given by:

$$\boldsymbol{\xi}^*(x) = -\frac{\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))}{||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V},$$

with

$$\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x)) = (I - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})(\nabla J(x))$$

1. A generic optimization algorithm

We can prove:

1. Constraints are asymptotically satisfied:

$$\boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x(0)) \text{ and } \boldsymbol{h}_{\widetilde{I}(x(t))} \leq e^{-\alpha_C t}\boldsymbol{h}(x(0))$$

2. J decreases as soon as the violation $\boldsymbol{C}_{\widetilde{I}(x(t))}$ is sufficiently small
3. All stationary points x^* of the ODE are KKT points

2. Applications to shape optimization

What is truely required by the user:

1. Specification of objective and constraints J, $\boldsymbol{g}$, $\boldsymbol{h}$

2. Applications to shape optimization

What is truely required by the user:

1. Specification of objective and constraints J, $\boldsymbol{g}$, $\boldsymbol{h}$
2. Fréchet derivatives $\mathrm{D}J(x)$, $\mathrm{D}\boldsymbol{g}(x)$, $\mathrm{D}\boldsymbol{h}(x)$ given as linear operators

2. Applications to shape optimization

What is truely required by the user:

1. Specification of objective and constraints J, $\boldsymbol{g}$, $\boldsymbol{h}$
2. Fréchet derivatives $\mathrm{D}J(x)$, $\mathrm{D}\boldsymbol{g}(x)$, $\mathrm{D}\boldsymbol{h}(x)$ given as linear operators
3. Scalar product a for identifying these derivatives

2. Applications to shape optimization

What is truely required by the user:

1. Specification of objective and constraints J, $\boldsymbol{g}$, $\boldsymbol{h}$
2. Fréchet derivatives $\mathrm{D}J(x)$, $\mathrm{D}\boldsymbol{g}(x)$, $\mathrm{D}\boldsymbol{h}(x)$ given as linear operators
3. Scalar product a for identifying these derivatives
4. Typical length scale Δt (e.g. the mesh size)

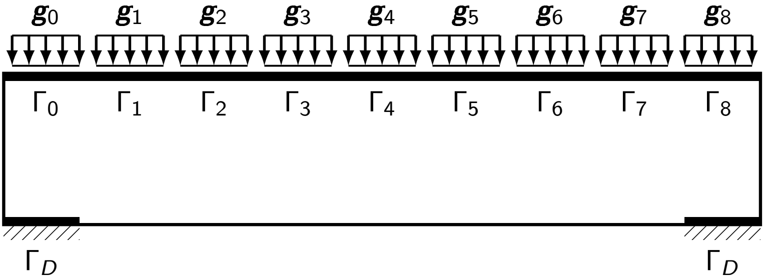
2. Applications to shape optimization

What is truely required by the user:

1. Specification of objective and constraints J, $\boldsymbol{g}$, $\boldsymbol{h}$
2. Fréchet derivatives $\mathrm{D}J(x)$, $\mathrm{D}\boldsymbol{g}(x)$, $\mathrm{D}\boldsymbol{h}(x)$ given as linear operators
3. Scalar product a for identifying these derivatives
4. Typical length scale Δt (e.g. the mesh size)
5. α_J and α_C for tuning the relative magnitude of $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$, i.e. the speed at which violated constraints become satisfied.

2. Applications to shape optimization

A multiple load case.



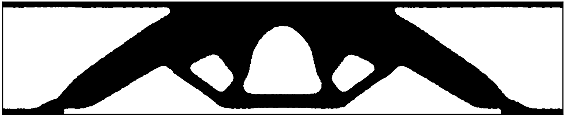
$$\begin{cases} -\mathrm{div}(Ae(\boldsymbol{u}_i)) = 0 & \text{in } \Omega \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = \boldsymbol{g}_i & \text{on } \Gamma_i \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma_j \text{ for } j \neq i \\ \boldsymbol{u}_i = 0 & \text{on } \Gamma_D, \end{cases}$$

2. Applications to shape optimization

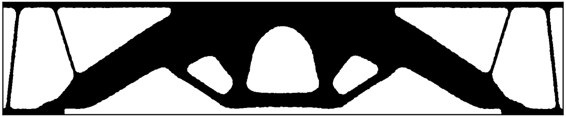
Volume minimization subject to multiple load rigidity constraints

$$\begin{aligned} \min_{\Omega} \quad & \int_{\Omega} \mathrm{d}x \\ s.t. & \int_{\Omega} Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\mathrm{d}x \leq C, \quad \forall i = 1 \ldots 9 \end{aligned}$$

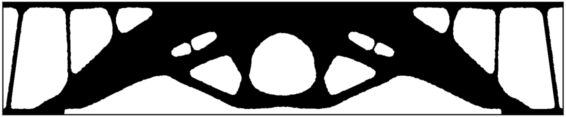
Demonstration on shape optimization test cases



(a) One load (only $\boldsymbol{g}_4$ is considered).

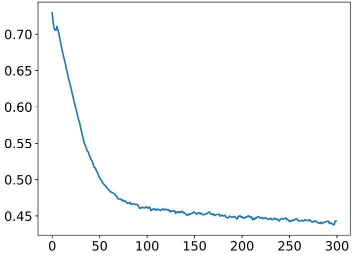


(b) Three loads (only $\boldsymbol{g}_0, \boldsymbol{g}_4, \boldsymbol{g}_8$ are considered).

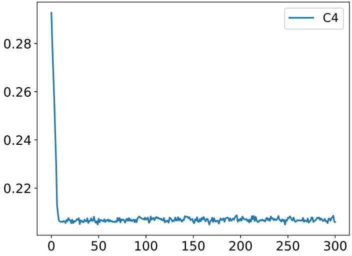


(c) All nine loads.

2. Applications to shape optimization



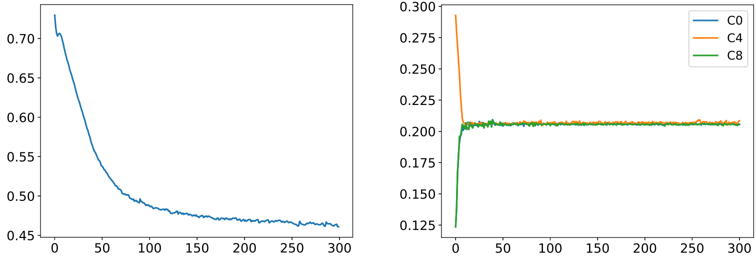
(a) $J(\Omega) = \mathrm{Vol}(\Omega)$.



(b) Constraints C_i.

Figure: Single load case.

2. Applications to shape optimization

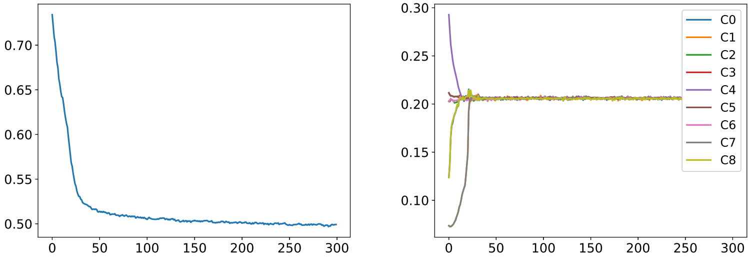


(a) $J(\Omega) = \mathrm{Vol}(\Omega)$.

(b) Constraints C_i.

Figure: Three load case.

2. Applications to shape optimization



(a) $J(\Omega) = \mathrm{Vol}(\Omega)$.

(b) Constraints C_i.

Figure: Nine load case.

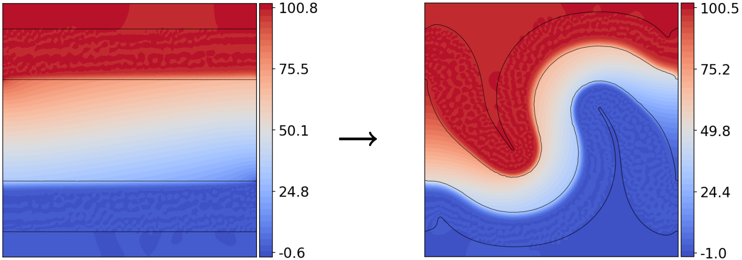
2. Applications to shape optimization

Heat exchange subject to maximal pressure drop and non penetration constraint:

$$\max_{\Omega} \quad \int_{\Omega_{f,cold}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x - \int_{\Omega_{f,hot}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x$$

$$s.t. \quad \int_{\partial\Omega_{f,out}} p \mathrm{d}s - \int_{\partial\Omega_{f,in}} p \mathrm{d}s \leq \mathrm{DP}_0,$$

$$d(\Omega_{f,hot}, \Omega_{f,cold}) \geqslant d_{\min}$$



References

- Feppon, F., Allaire, G., Bordeu, F., Cortial, J., and Dapogny, C. Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework.
  SeMA Journal (2019).
- Feppon, F., Allaire, G., and Dapogny, C. Null space gradient flows for constrained optimization with applications to shape optimization.
  HAL preprint hal-01972915 (2019).
- Feppon, F., Allaire, G., and Dapogny, C. A variational formulation for computing shape derivatives of geometric constraints along rays.
  HAL preprint hal-01879571 (2019).

Many thanks!

Constrained optimization

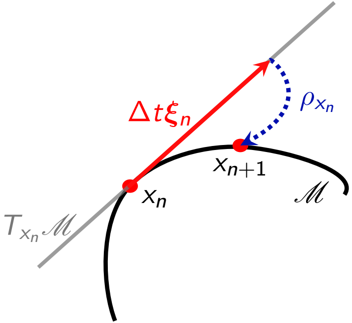
- For a vector space $\mathcal{X} = V$, a sequence of updates will be of the form

$$x_{n+1} = x_n - \Delta t \boldsymbol{\xi}_n$$

where $-\boldsymbol{\xi}_n$ is the current descent direction.

- For a manifold, this becomes

$$x_{n+1} = \rho_{x_n}(-\Delta t \boldsymbol{\xi}_n)$$



1. A generic optimization algorithm

Warning: $\nabla J(x)$ and the transpose $^{\mathcal{T}}$ must be computed with respect to the scalar product a of the Hilbert space V or T_{x_n}. In practice this means solving

$$\forall \boldsymbol{\xi} \in V,\ a(\nabla J(x), \boldsymbol{\xi}) = \mathrm{D}J(x)\boldsymbol{\xi}$$

$$\forall \boldsymbol{\xi} \in V,\ a(\nabla g_i(x), \boldsymbol{\xi}) = \mathrm{D}g_i(x)\boldsymbol{\xi}$$

$$\forall \boldsymbol{\xi} \in V,\ a(\nabla h_i(x), \boldsymbol{\xi}) = \mathrm{D}h_i(x)\boldsymbol{\xi}$$

Then

$$\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(x) = \begin{bmatrix} \nabla g_0(x) & \cdots & \nabla g_p(x) \end{bmatrix}^T$$

$$\mathrm{D}\boldsymbol{h}^{\mathcal{T}}(x) = \begin{bmatrix} \nabla h_0(x) & \cdots & \nabla h_q(x) \end{bmatrix}^T$$