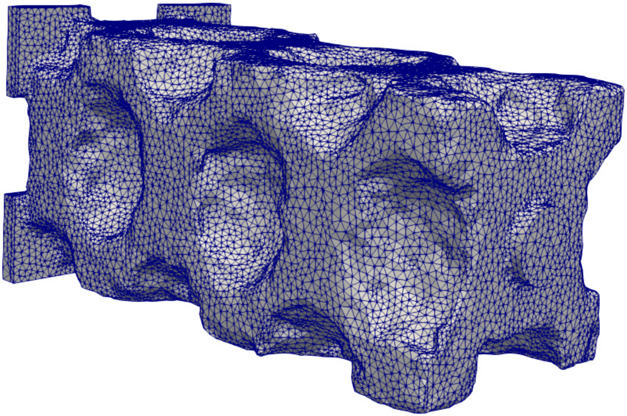
Parallel three-dimensional topology optimization of multiphysics systems

Florian Feppon

Grégoire Allaire – Charles Dapogny
Julien Cortial – Felipe Bordeu.

WCCM-ECCOMAS 2020

Body-fitted topology optimization

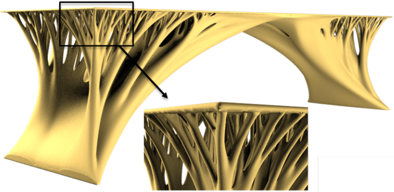


Topology optimisation of a bending 3D cantilever beam

State of the art for 3D optimization

(a) Frustum (2017)

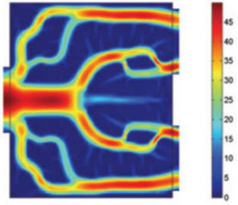
(b) APWorks (2016)



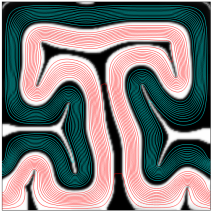
(c) M2DO (Kambampati et. al. 2018)

State of the art for 3D optimization

For thermal-fluid systems it is still an active research field.

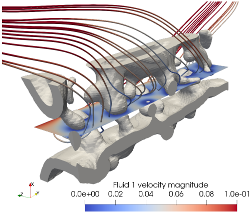


(a) Dede (2009, Toyota)



(b) Papazoglou (2015, TU Delft)

(c) Savier (2019, United Technologies)



(d) Hoghoj et. al (2020, DTU)

Fluid pipes optimized for convective heat transfer.

Outline

1. Multiphysics model

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives
   2.2 Null space optimization algorithm

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
    2.1 Body-fitted topology optimization with shape derivatives
    2.2 Null space optimization algorithm
    2.3 Parallel computing

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
    2.1 Body-fitted topology optimization with shape derivatives
    2.2 Null space optimization algorithm
    2.3 Parallel computing
3. 3D test cases

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives
   2.2 Null space optimization algorithm
   2.3 Parallel computing
3. 3D test cases
   3.1 Lift-Drag optimization

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives
   2.2 Null space optimization algorithm
   2.3 Parallel computing
3. 3D test cases
   3.1 Lift-Drag optimization
   3.2 Fluid-Structure interaction

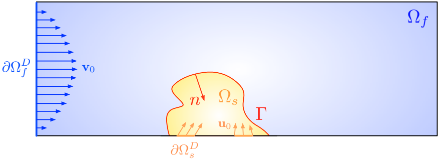
Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives
   2.2 Null space optimization algorithm
   2.3 Parallel computing
3. 3D test cases
   3.1 Lift-Drag optimization
   3.2 Fluid-Structure interaction
   3.3 Convective heat transfer

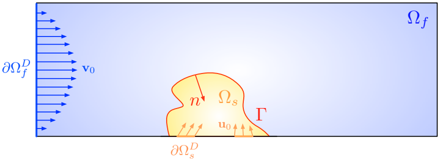
Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
   2.1 Body-fitted topology optimization with shape derivatives
   2.2 Null space optimization algorithm
   2.3 Parallel computing
3. 3D test cases
   3.1 Lift-Drag optimization
   3.2 Fluid-Structure interaction
   3.3 Convective heat transfer
   3.4 Fluid-to-fluid heat exchangers

1. Multiphysics model



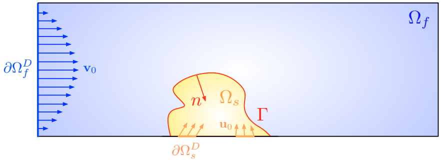
1. Multiphysics model



- Incompressible Navier-Stokes equations for $(\boldsymbol{v}, p)$ in Ω_f

$$-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = \boldsymbol{f}_f \text{ in } \Omega_f$$

1. Multiphysics model



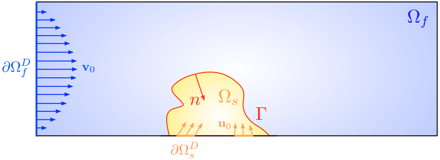
- Incompressible Navier-Stokes equations for $(\boldsymbol{v}, p)$ in Ω_f

$$-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = \boldsymbol{f}_f \text{ in } \Omega_f$$

- Steady-state convection-diffusion for T_f and T_s in Ω_f and Ω_s:

$$\begin{aligned} -\mathrm{div}(k_f \nabla T_f) + \rho \boldsymbol{v} \cdot \nabla T_f &= Q_f \quad \text{in } \Omega_f \\ -\mathrm{div}(k_s \nabla T_s) &= Q_s \quad \text{in } \Omega_s \end{aligned}$$

1. Multiphysics model



- Incompressible Navier-Stokes equations for $(\boldsymbol{v}, p)$ in Ω_f

$$-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = \boldsymbol{f}_f \text{ in } \Omega_f$$

- Steady-state convection-diffusion for T_f and T_s in Ω_f and Ω_s:

$$\begin{aligned} -\mathrm{div}(k_f \nabla T_f) + \rho \boldsymbol{v} \cdot \nabla T_f &= Q_f \quad \text{in } \Omega_f \\ -\mathrm{div}(k_s \nabla T_s) &= Q_s \quad \text{in } \Omega_s \end{aligned}$$

- Linearized thermoelasticity with fluid-structure interaction for $\boldsymbol{u}$ in Ω_s:

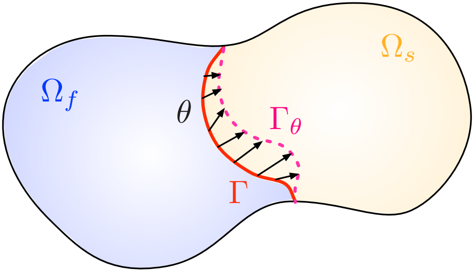
$$\begin{aligned} -\mathrm{div}(\sigma_s(\boldsymbol{u}, T_s)) &= \boldsymbol{f}_s \quad && \text{in } \Omega_s \\ \sigma_s(\boldsymbol{u}, T_s) \cdot \boldsymbol{n} &= \sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{n} && \text{on } \Gamma. \end{aligned}$$

Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients

2. Numerical methodology

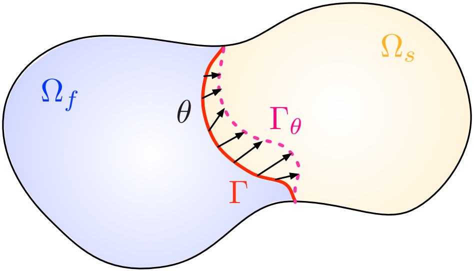
Hadamard's boundary variation method



$$\min_{\Gamma} J(\Gamma)$$

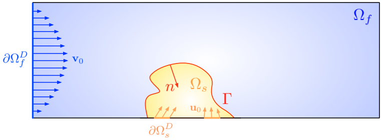
2. Numerical methodology

Hadamard's boundary variation method



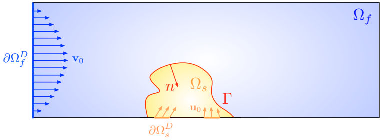
$$\min_{\Gamma} J(\Gamma)$$

$$J(\Gamma_{\boldsymbol{\theta}}) = J(\Gamma) + \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}}(\boldsymbol{\theta}) + o(\boldsymbol{\theta})$$



Proposition

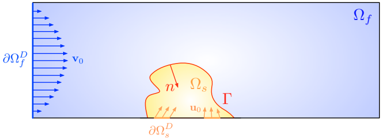
Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ *an arbitrary functional with continuous partial derivatives and* $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ *the above state variables. Then, if these are smooth enough,* $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ *is shape differentiable and the derivative reads:*



Proposition

Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta}) \\
&\quad= \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q) \nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p) \nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad+ \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}} \right) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad+ \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r}) \nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s) \nabla \boldsymbol{r} \cdot \boldsymbol{n}) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s
\end{aligned}
$$



Proposition

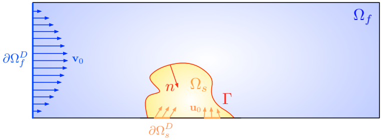
Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta})$$

$$= \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q)\nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p)\nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$

$$+ \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}} \right) (\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$

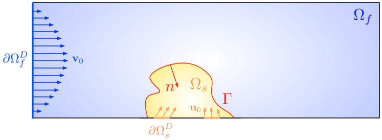
$$+ \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r})\nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s)\nabla \boldsymbol{r} \cdot \boldsymbol{n})\,(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$



Proposition

Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta}) \\
&\quad = \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q) \nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p) \nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s \\
&\quad + \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}}\right)(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s \\
&\quad + \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r}) \nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s) \nabla \boldsymbol{r} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s
\end{aligned}$$



Proposition

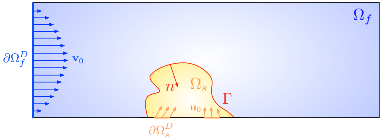
Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta})$$

$$= \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q)\nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p)\nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$

$$+ \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}}\frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}}\frac{\partial S_f}{\partial \boldsymbol{n}}\right)(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$

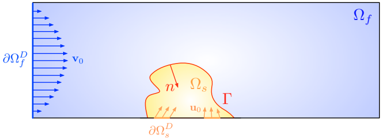
$$+ \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r})\nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s)\nabla \boldsymbol{r} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n})\mathrm{d}s$$



Proposition

Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta}) \\
&\quad = \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q) \nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p) \nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad + \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}} \right) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad + \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r}) \nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s) \nabla \boldsymbol{r} \cdot \boldsymbol{n}) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s
\end{aligned}$$



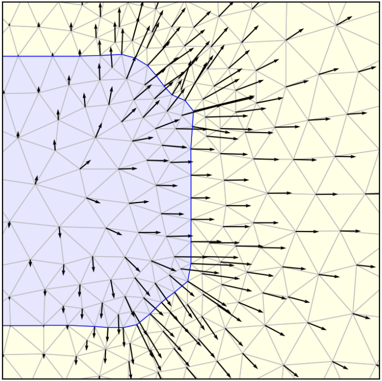
Proposition

Let $J(\Gamma, \boldsymbol{u}, T, \boldsymbol{v}, p)$ an arbitrary functional with continuous partial derivatives and $\boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma)$ the above state variables. Then, if these are smooth enough, $\Gamma \mapsto J(\Gamma, \boldsymbol{u}(\Gamma), T(\Gamma), \boldsymbol{v}(\Gamma), p(\Gamma))$ is shape differentiable and the derivative reads:

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\Big[J(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\Big](\boldsymbol{\theta}) \\
&\quad = \overline{\frac{\partial \mathfrak{J}}{\partial \boldsymbol{\theta}}}(\boldsymbol{\theta}) + \int_\Gamma (\boldsymbol{f}_f \cdot \boldsymbol{w} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{w}, q) \nabla \boldsymbol{v} \cdot \boldsymbol{n} + \boldsymbol{n} \cdot \sigma_f(\boldsymbol{v}, p) \nabla \boldsymbol{w} \cdot \boldsymbol{n})(\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad + \int_\Gamma \left(k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f + Q_f S_f - Q_s S_s - 2k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}} \right) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s \\
&\quad + \int_\Gamma (\sigma_s(\boldsymbol{u}, T_s) : \nabla \boldsymbol{r} - \boldsymbol{f}_s \cdot \boldsymbol{r} - \boldsymbol{n} \cdot Ae(\boldsymbol{r}) \nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \sigma_s(\boldsymbol{u}, T_s) \nabla \boldsymbol{r} \cdot \boldsymbol{n}) (\boldsymbol{\theta} \cdot \boldsymbol{n}) \mathrm{d}s
\end{aligned}
$$

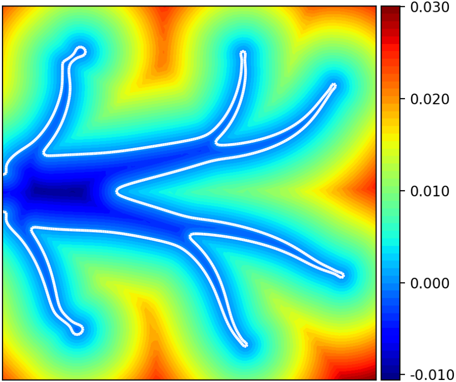
We rely on the algorithm proposed by Allaire, Dapogny, Frey (2013):

1. Given a mesh and a moving vector field



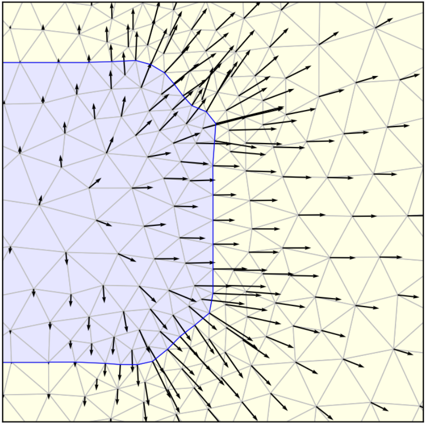
We consider the algorithm proposed by Allaire, Dapogny, Frey (2013):

2. A level-set function ϕ associated to $\Omega_f \subset D$ is computed on the mesh.



We consider the algorithm proposed by Allaire, Dapogny, Frey (2013):

3. The level-set function is avected on the computational domain which is then adaptively remeshed:

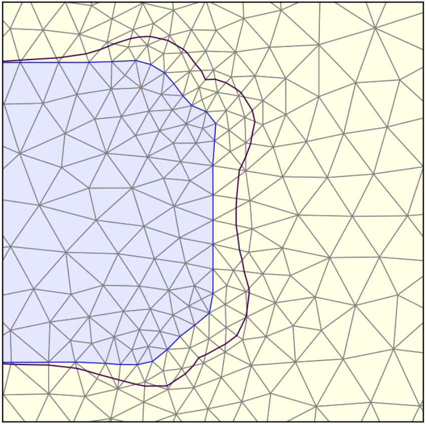


We consider the algorithm proposed by Allaire, Dapogny, Frey (2013):

3. The level-set function is avected on the computational domain which is then adaptively remeshed:

Advection of a level set for Ω_f on the computational mesh.

$$\partial_t \phi + \boldsymbol{\theta} \cdot \nabla \phi = 0$$



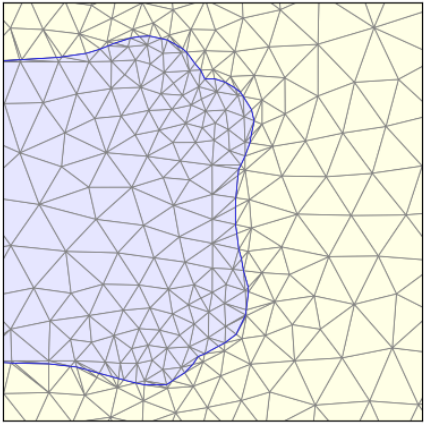
2. Numerical methodology

Level set based mesh evolution method

We consider the algorithm proposed by Allaire, Dapogny, Frey (2013):

3. The level-set function is avected on the computational domain which is then adaptively remeshed:

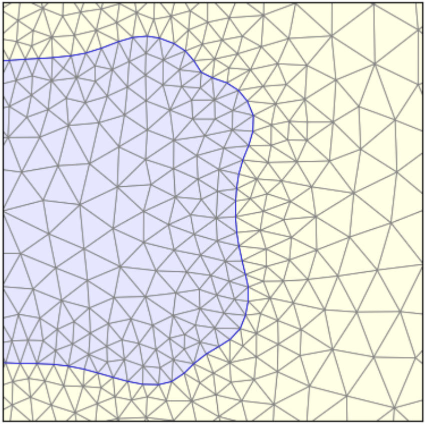
Breaking the zero isoline of the level set (with the `mmg` library).



We consider the algorithm proposed by Allaire, Dapogny, Frey (2013):

3. The level-set function is avected on the computational domain which is then adaptively remeshed:

Remeshing adaptively the computational mesh (with `mmg` library).



Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
    2.1 Body-fitted topology optimization with shape derivatives
    2.2 Null space optimization algorithm

$$\min_{\Gamma} \quad J(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma))$$

$$s.c. \begin{cases} g_i(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma)) = 0, & 1 \leq i \leq p, \\ h_j(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma)) \leq 0, & 1 \leq j \leq q, \end{cases}$$

J : objective function

g_i : equality constraints

h_j : inequality constraints

We solve generic optimization problem with the "null space" gradient flow:

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

with

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

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

We solve generic optimization problem with the “null space” gradient flow:

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

with

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

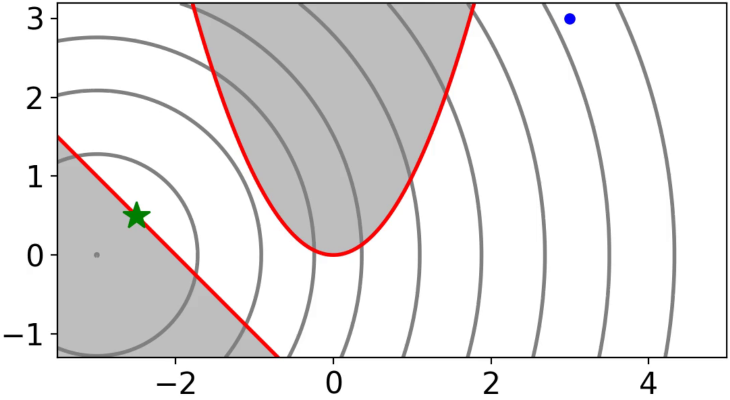
$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T (\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)} \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T)^{-1} \boldsymbol{C}_{\widetilde{I}(x)}(x).$$

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

Outline

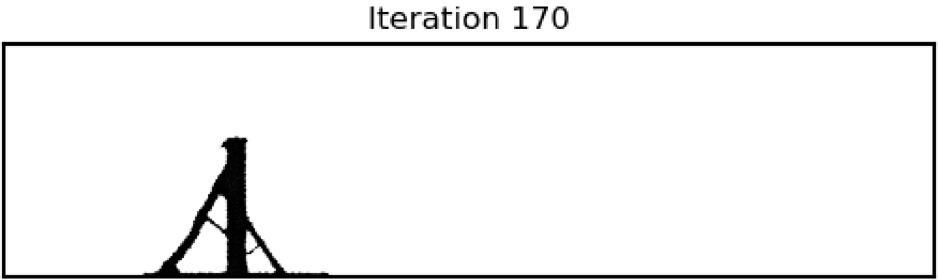
$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad J(x_1,x_2) = x_1^2 + (x_2+3)^2$$

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



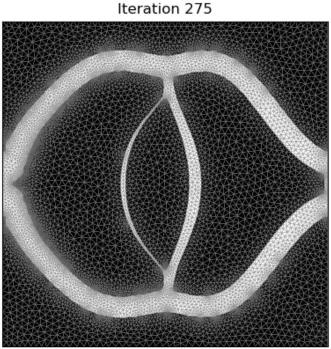
Outline

Interlude: 2D fluid-structure interaction



Outline

Interlude: 2D convective heat transfer



Outline

1. Multiphysics model
2. Numerical methodology: three main ingredients
    2.1 Body-fitted topology optimization with shape derivatives
    2.2 Null space optimization algorithm
    2.3 Parallel computing

- In theory, exact same methodology: same shape derivative formulas, same optimization algorithm.

- In theory, exact same methodology: same shape derivative formulas, same optimization algorithm.
- In practice, requires a completely revised implementation due to:

- In theory, exact same methodology: same shape derivative formulas, same optimization algorithm.
- In practice, requires a completely revised implementation due to:
  - large size of finite element linear systems: $\longrightarrow$ preconditioning + domain decomposition;

2. Numerical methodology

Parallel computing and 3D implementation

- In theory, exact same methodology: same shape derivative formulas, same optimization algorithm.
- In practice, requires a completely revised implementation due to:
  - large size of finite element linear systems: $\longrightarrow$ preconditioning $+$ domain decomposition;
  - 3D remeshing.

For domain decomposition, we use PETSc through the FreeFEM interface of P. Jolivet. We use the following physically dependent Domain Decomposition preconditioners:

- Linear elasticity : Geometric Algebraic Multigrid (GAMG) + CG

[1] Johann Moulin, Pierre Jolivet, and Olivier Marquet. "Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis". In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 718–743.

For domain decomposition, we use PETSc through the FreeFEM interface of P. Jolivet. We use the following physically dependent Domain Decomposition preconditioners:

- Linear elasticity : Geometric Algebraic Multigrid (GAMG) + CG
- Thermal conduction and convection: hypre

[1] Johann Moulin, Pierre Jolivet, and Olivier Marquet. "Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis". In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 718–743.

For domain decomposition, we use PETSc through the FreeFEM interface of P. Jolivet. We use the following physically dependent Domain Decomposition preconditioners:

- Linear elasticity : Geometric Algebraic Multigrid (GAMG) + CG
- Thermal conduction and convection: hypre
- Navier-Stokes equation with the Newton method: Augmented Lagrangian preconditioner for the Oseen problem[1].

[1] Johann Moulin, Pierre Jolivet, and Olivier Marquet. "Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis". In: *Computer Methods in Applied Mechanics and Engineering* 351 (2019), pp. 718–743.

Outline

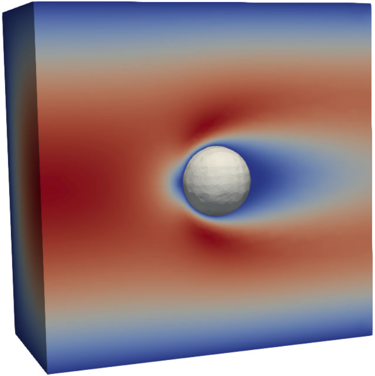
1. Multiphysics model
2. Numerical methodology: three main ingredients
    2.1 Body-fitted topology optimization with shape derivatives
    2.2 Null space optimization algorithm
    2.3 Parallel computing
3. 3D test cases

$$\min \quad -\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

$$s.c. \begin{cases} \mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \mathrm{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$

$$\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_{\Gamma} \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

$$\mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$



3. 3D cases

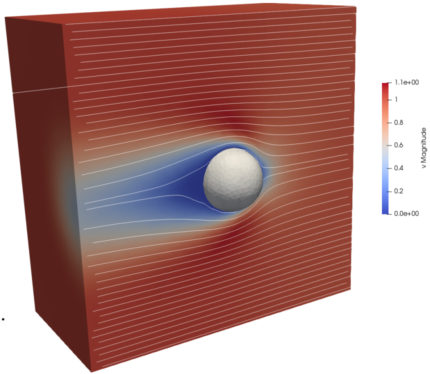
Lift-Drag optimization

$$\min \quad -\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

$$s.c. \begin{cases} \operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \operatorname{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$

$$\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

$$\operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$

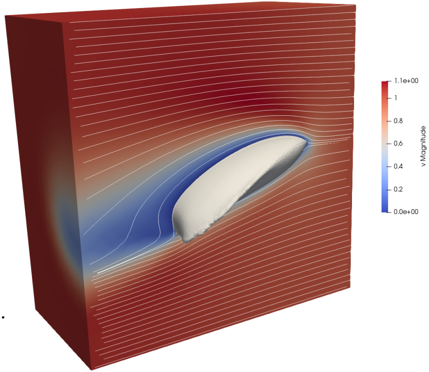


$$
\min \quad -\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))
$$

$$
s.c. \begin{cases} \operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \operatorname{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}
$$

$$
\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_{\Gamma} \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,
$$

$$
\operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.
$$

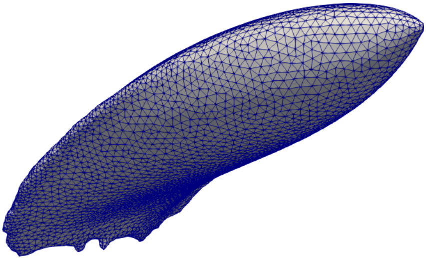


3. 3D cases

Lift-Drag optimization

$$\min \quad -\,\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

$$s.c. \begin{cases} \mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \mathrm{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$



$$\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p)\boldsymbol{n} \mathrm{d}s,$$

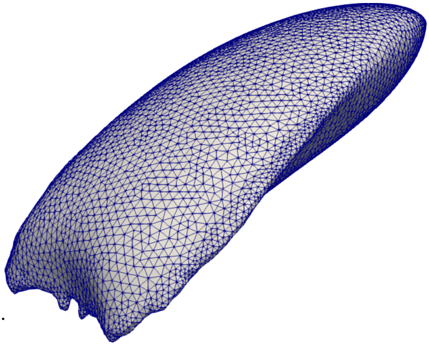
$$\mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$

$$\min \quad -\,\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

$$s.c. \begin{cases} \mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \mathrm{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x}\mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$

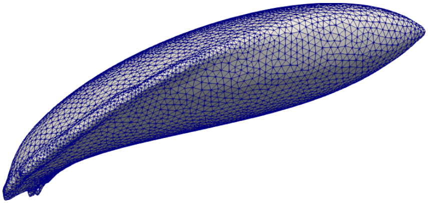
$$\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p)\boldsymbol{n}\mathrm{d}s,$$

$$\mathrm{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v}\mathrm{d}x.$$



$$\min \quad -\text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

$$s.c. \begin{cases} \text{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \texttt{DRAG}_0 \\ \text{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$



$$\text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

$$\text{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$

3. 3D cases

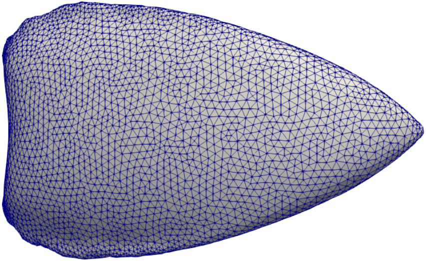
Lift-Drag optimization

$$\min \quad -\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$$

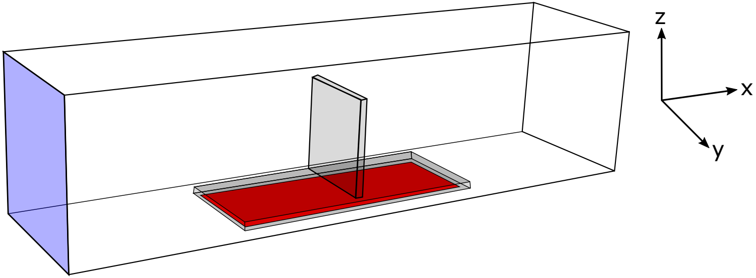
$$s.c. \begin{cases} \operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \mathtt{DRAG}_0 \\ \operatorname{Vol}(\Omega_f) = V_0 \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x} \mathrm{d}x = \boldsymbol{x}_0, \end{cases}$$

$$\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

$$\operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$

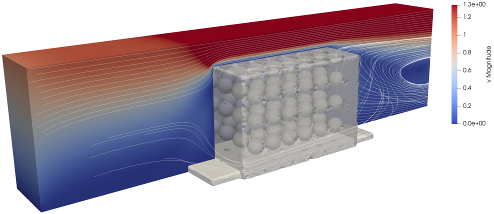


Minimization of the rigidity of a supporting structure subject to the pressure of an incoming flow.

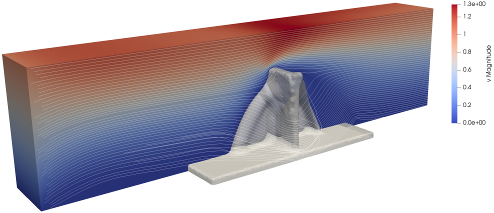


$$\min \quad \int_{\Omega_s} Ae(\boldsymbol{u}) : e(\boldsymbol{u}) \mathrm{d}x$$
$$s.c. \quad \mathrm{Vol}(\Omega_s) = \mathrm{Vol}_{target}.$$

Minimization of the rigidity of a supporting structure subject to the pressure of an incoming flow.



(a) Initial shape

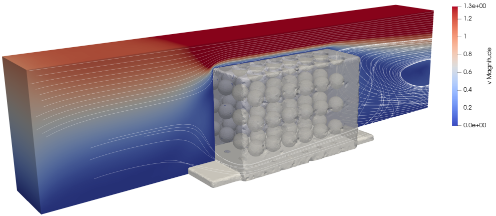


(b) Optimized shape

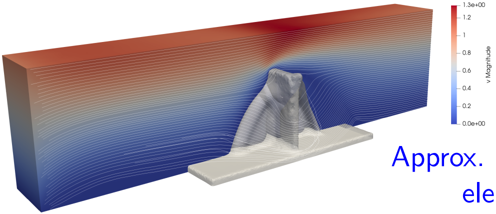
3. 3D cases

Interaction fluide-structure

Minimization of the rigidity of a supporting structure subject to the pressure of an incoming flow.

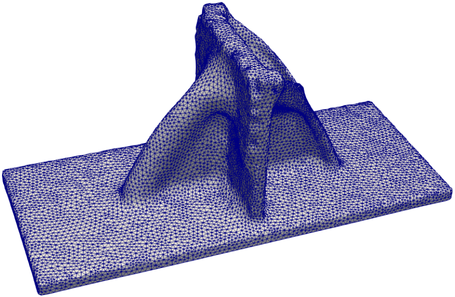


(a) Initial shape

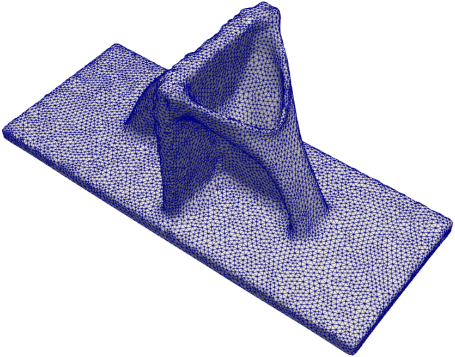


Approx. 2 millions of elements.

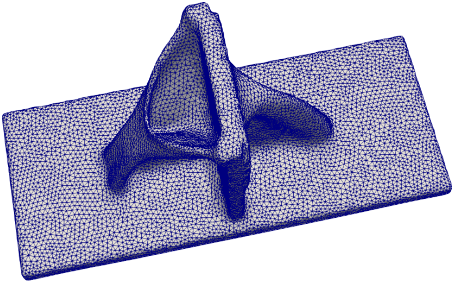
(b) Optimized shape



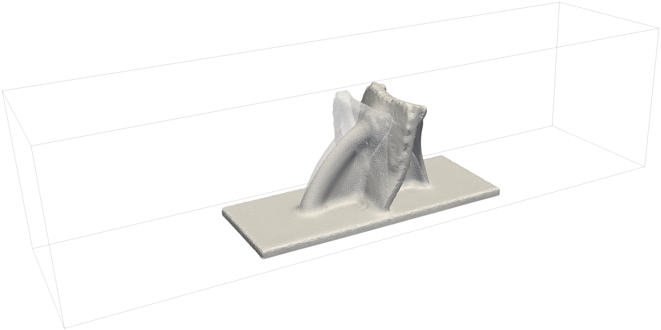
Optimized shape.



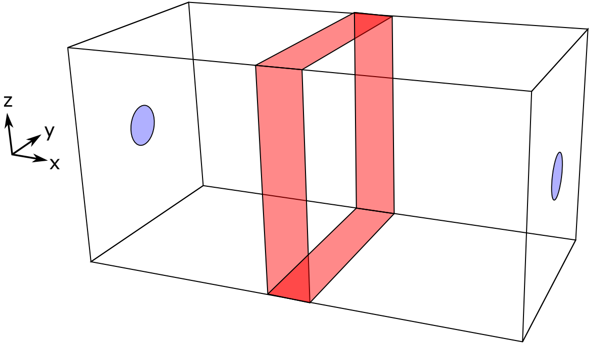
Optimized shape.



Optimized shape.



Elastic deformation.

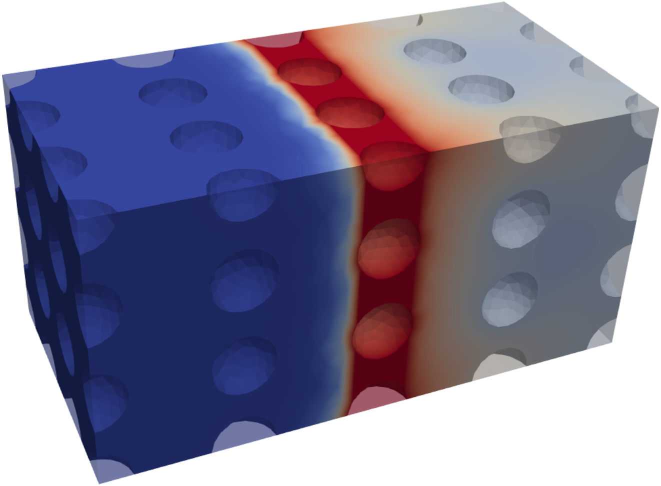


$$\min_{\Gamma} \quad J(\Gamma, \boldsymbol{v}(\Gamma), T(\Gamma)) := -\int_{\Omega_f} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x$$

$$\text{s.t.} \quad \begin{cases} \mathrm{DP}(p(\Gamma)) := \int_{\partial\Omega_f^{\text{in}}} p \mathrm{d}s - \int_{\partial\Omega_f^N} p \mathrm{d}s \leqslant \mathrm{DP}_T \\ \mathrm{Vol}(\Omega_f) = V_T. \end{cases}$$

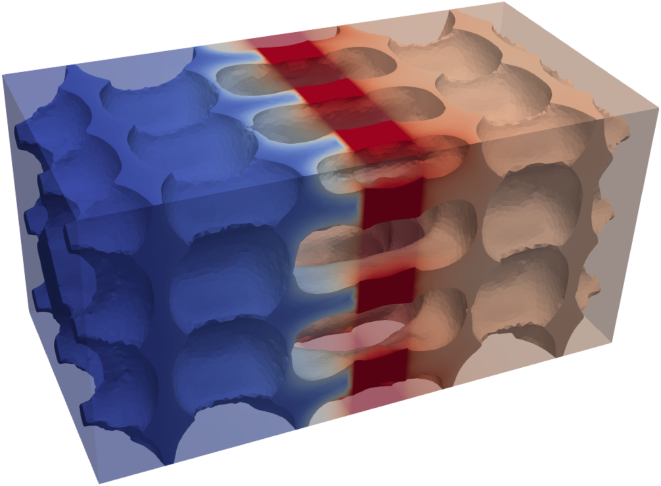
3. 3D cases

3D convective heat transfer



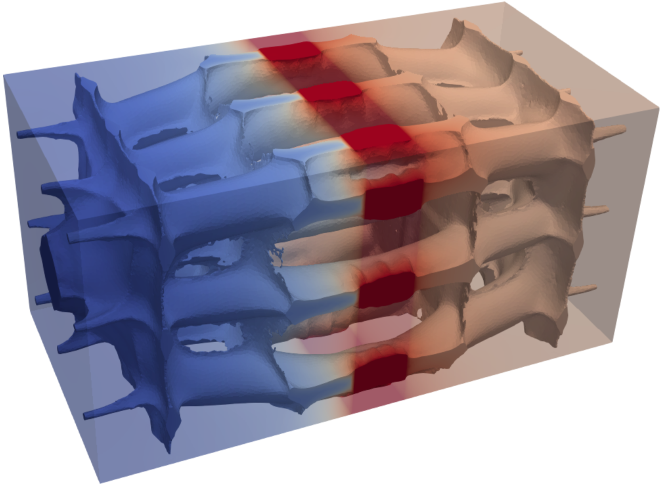
3. 3D cases

3D convective heat transfer



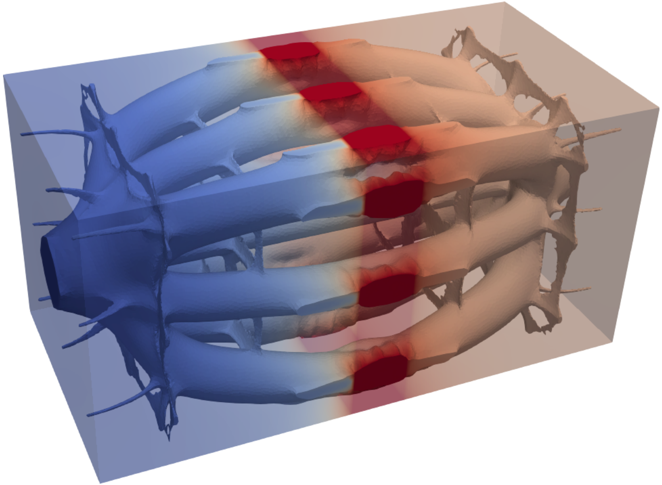
3. 3D cases

3D convective heat transfer



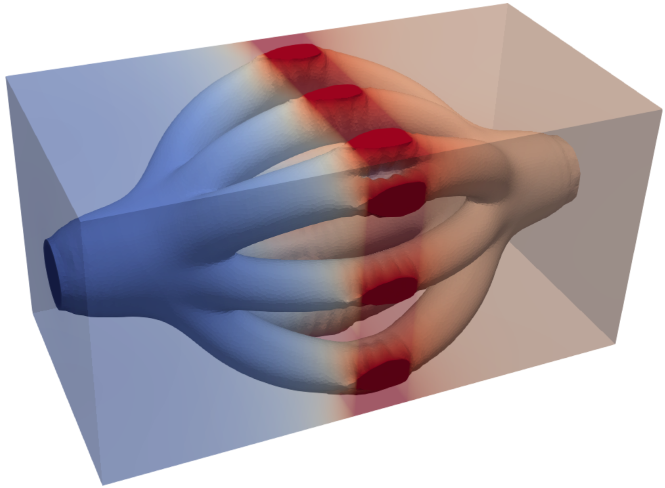
3. 3D cases

3D convective heat transfer



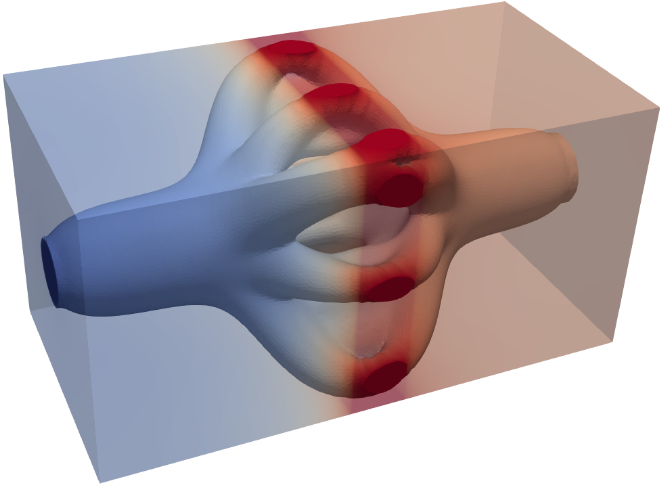
3. 3D cases

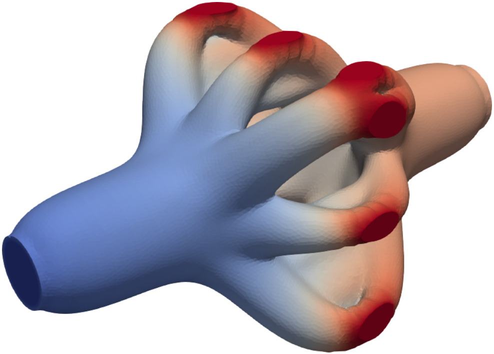
3D convective heat transfer



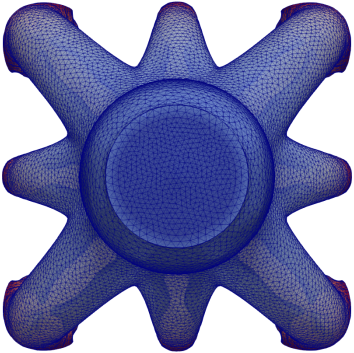
3. 3D cases

3D convective heat transfer

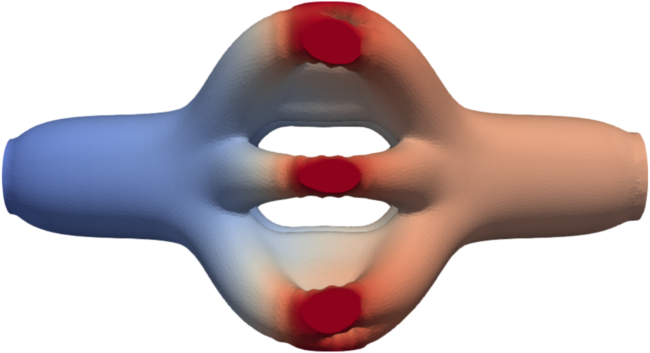




Optimized design.



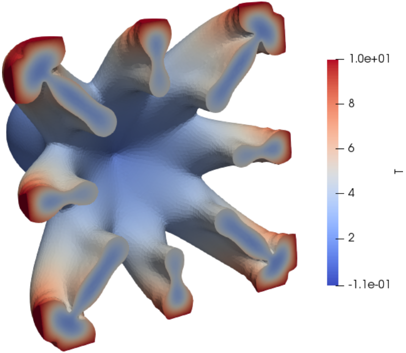
Optimized design.



Optimized design.

3. 3D cases

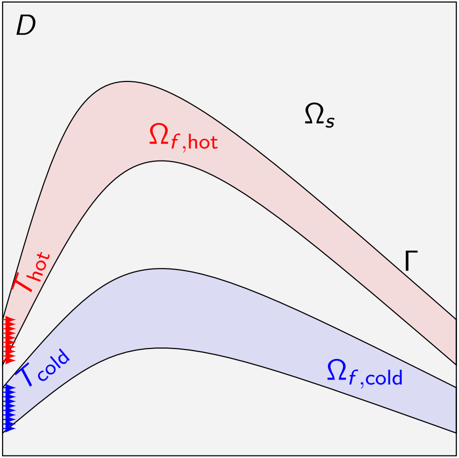
3D convective heat transfer



Optimized design.

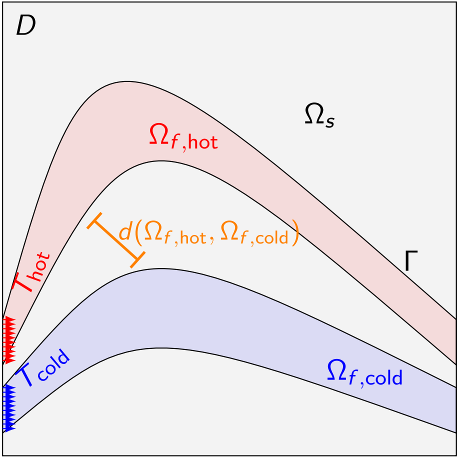
Outline

Heat exchangers



Settings of the heat exchanger topology optimization problem .

Heat exchangers



Settings of the heat exchanger topology optimization problem .

Non-penetration constraint:

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

Heat exchangers

Fluid-to-fluid heat exchangers

$$\min_{\Omega_f \subset D} \quad J(\Omega_f) = -\left(\int_{\Omega_{f,\text{cold}}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x - \int_{\Omega_{f,\text{hot}}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x \right)$$

$$\text{s.t.} \begin{cases} \mathrm{Vol}(\Omega_{f,\text{hot}}) := \int_{\Omega_{f,\text{hot}}} \mathrm{d}x \leq \mathtt{V}_0 \\ \mathrm{Vol}(\Omega_{f,\text{cold}}) := \int_{\Omega_{f,\text{cold}}} \mathrm{d}x \leq \mathtt{V}_0 \\ \mathtt{DP}(\Omega_{f,\text{hot}}) = \int_{\partial \Omega^D_{f,\text{hot}}} p \mathrm{d}s - \int_{\partial \Omega^N_{f,\text{hot}}} p \mathrm{d}s \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\text{cold}}) = \int_{\partial \Omega^D_{f,\text{cold}}} p \mathrm{d}s - \int_{\partial \Omega^N_{f,\text{cold}}} p \mathrm{d}s \leq \mathtt{DP}_0 \\ P_{\text{hot}\to\text{cold}}(\Omega_f) \geqslant d_{\min} \\ P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}, \end{cases}$$

Heat exchangers

Fluid-to-fluid heat exchangers

$$\min_{\Omega_f \subset D} \quad J(\Omega_f) = -\left(\int_{\Omega_{f,\text{cold}}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x - \int_{\Omega_{f,\text{hot}}} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x \right)$$

$$\text{s.t.} \begin{cases} \mathrm{Vol}(\Omega_{f,\text{hot}}) := \int_{\Omega_{f,\text{hot}}} \mathrm{d}x \leq \mathtt{V}_0 \\ \mathrm{Vol}(\Omega_{f,\text{cold}}) := \int_{\Omega_{f,\text{cold}}} \mathrm{d}x \leq \mathtt{V}_0 \\ \mathtt{DP}(\Omega_{f,\text{hot}}) = \int_{\partial\Omega^D_{f,\text{hot}}} p \mathrm{d}s - \int_{\partial\Omega^N_{f,\text{hot}}} p \mathrm{d}s \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\text{cold}}) = \int_{\partial\Omega^D_{f,\text{cold}}} p \mathrm{d}s - \int_{\partial\Omega^N_{f,\text{cold}}} p \mathrm{d}s \leq \mathtt{DP}_0 \\ P_{\text{hot}\to\text{cold}}(\Omega_f) \geqslant d_{\min} \\ P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}, \end{cases}$$

Difficulty: non-mixing constraint.

Heat exchangers

Non-mixing constraint

We enforce a minimum distance constraint between the two phases

$$\forall x \in \Omega_{f,\text{cold}}, \;\; d_{\Omega_{f,\text{hot}}}(x) \geqslant d_{\min},$$

by means of an approximate penalty functional

$$P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}$$

Heat exchangers

Non-mixing constraint

We enforce a minimum distance constraint between the two phases

$$\forall x \in \Omega_{f,\text{cold}}, \;\; d_{\Omega_{f,\text{hot}}}(x) \geqslant d_{\min},$$

by means of an approximate penalty functional

$$P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}$$

where

$$P_{\text{cold}\to\text{hot}}(\Omega_f) := \left\| \frac{1}{d_{\Omega_{f,\text{hot}}}} \right\|_{L^p(\partial\Omega_{f,\text{cold}})}^{-1} = \left(\int_{\partial\Omega_{f,\text{cold}}} \frac{1}{|d_{\Omega_{f,\text{hot}}}|^p} \mathrm{d}s \right)^{-\frac{1}{p}}.$$

Heat exchangers
Non-mixing constraint

We enforce a minimum distance constraint between the two phases

$$\forall x \in \Omega_{f,\text{cold}}, \quad d_{\Omega_{f,\text{hot}}}(x) \geqslant d_{\min},$$

by means of an approximate penalty functional

$$P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}$$

where

$$P_{\text{cold}\to\text{hot}}(\Omega_f) := \left|\left| \frac{1}{d_{\Omega_{f,\text{hot}}}} \right|\right|^{-1}_{L^p(\partial\Omega_{f,\text{cold}})} = \left(\int_{\partial\Omega_{f,\text{cold}}} \frac{1}{|d_{\Omega_{f,\text{hot}}}|^p} \mathrm{d}s \right)^{-\frac{1}{p}}.$$

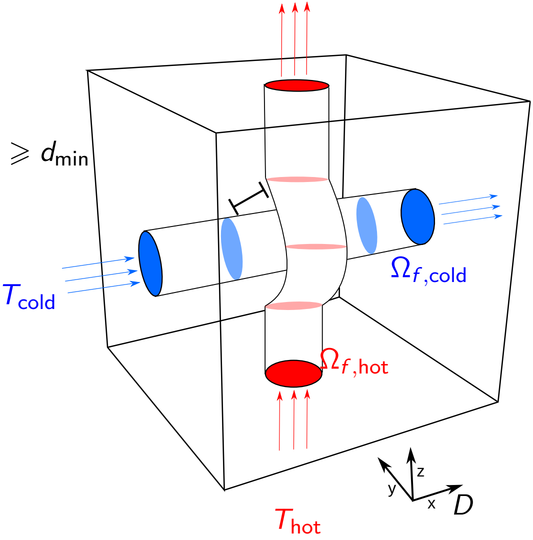
The shape derivative of $P_{\text{cold}\to\text{hot}}$ involving $d_{\Omega_{f,\text{hot}}}$ is computed conveniently with the variational method of

Florian Feppon, Grégoire Allaire, and Charles Dapogny. ''A variational formulation for computing shape derivatives of geometric constraints along rays''. In: *ESAIM: M2AN* 54.1 (2020), pp. 181{228.

Heat exchangers

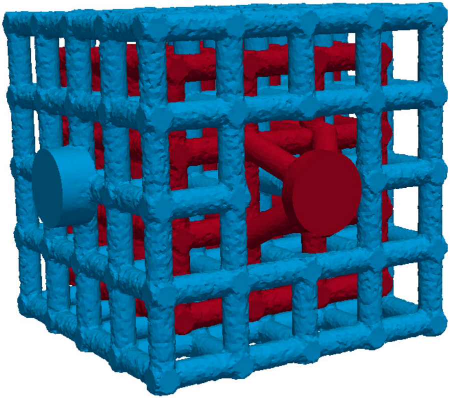
Fluid-to-fluid heat exchangers

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

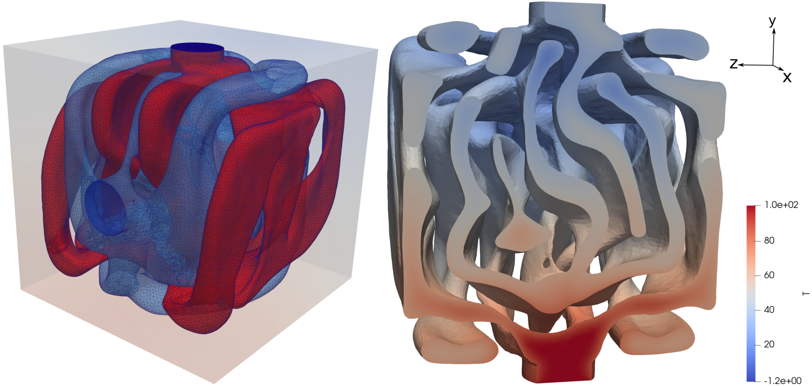


Heat exchangers

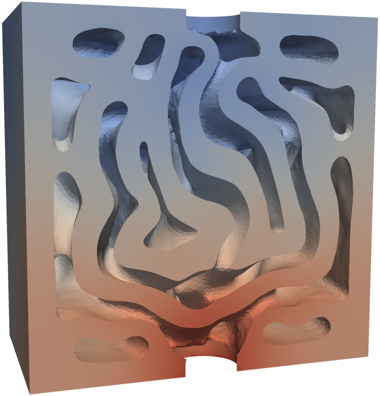
Fluid-to-fluid heat exchangers

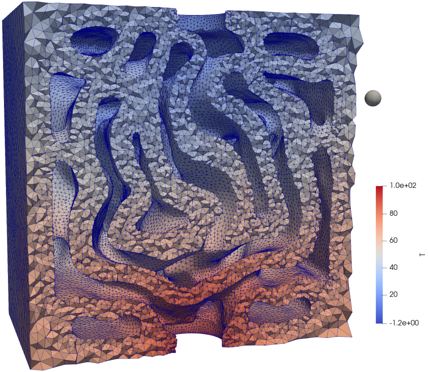


Heat exchangers



Heat exchangers





Thanks for your attention

F. Feppon et al. "Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing". In: *Journal of Computational Physics* 417 (2020), p. 109574

F Feppon et al. "Body-fitted topology optimization of 2D and 3D fluid-to-fluid heat exchangers". Aug. 2020