Shape and Topology optimization applied to Compact Heat Exchangers

Florian Feppon

PhD advisors : Grégoire Allaire, Charles Dapogny

Safran Tech advisors : Julien Cortial, Felipe Bordeu.

Von Karmann Institute
Compact Heat Exchangers in Additive Manufacturing – 2021, April 27th

Topology optimization

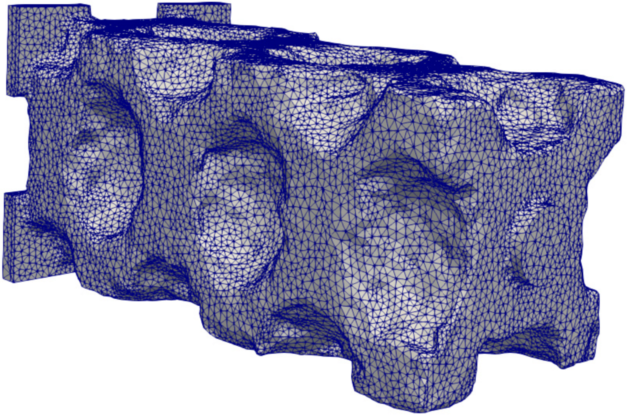
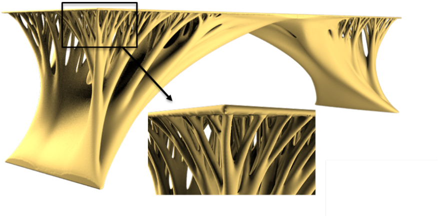


Figure: Optimization of the rigidity of a mechanical structure subject to flexural load

State of the art for 3D topology optimization

(a) Siemens (2017)

(b) APWorks (2016)

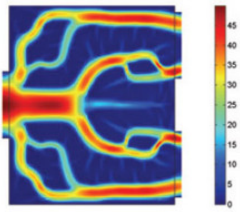


(c) M2DO (Kambampati et. al. 2018)

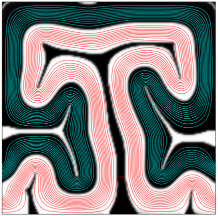
(d) AIRBUS (2010)

State of the art for 3D topology optimization

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



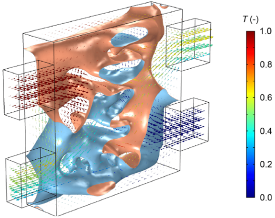
(a) Dede (2009, Toyota)



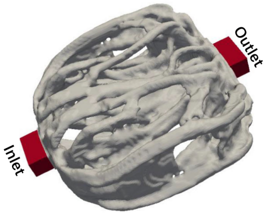
(b) Papazoglou (2015, TU Delft)

Figure: Fluid pipes optimized for convective heat transfer.

State of the art for 3D topology optimization

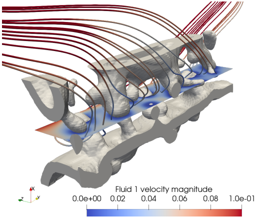


(a) Kobayashi et. al. (2020)



(b) Yu et. al. (2020)

(c) Savier (2019, United Technologies)



(d) Hoghoj et. al. (2020)

Figure: Fluid pipes optimized for convective heat transfer with density methods.

State of the art for 3D topology optimization

The objective today: shape and topologically optimized heat exchangers with the method of Hadamard and body-fitted meshes.

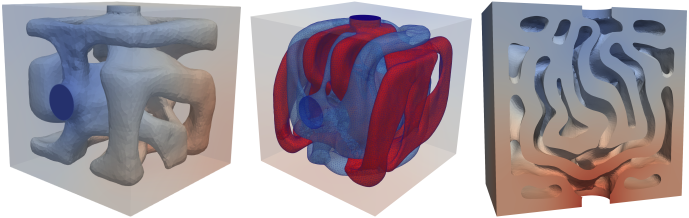
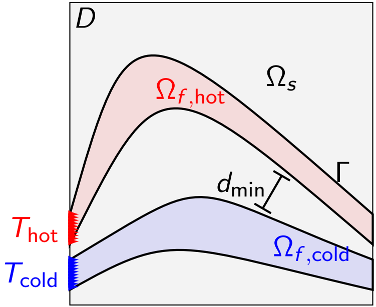


Figure: Topology optimized heat exchanger devices with the method of Hadamard and a body-fitted mesh evolution algorithm. Figures from [1].

[1] Feppon et al., *Body-fitted topology optimization of 2D and 3D fluid-to-fluid heat exchangers* (2021)

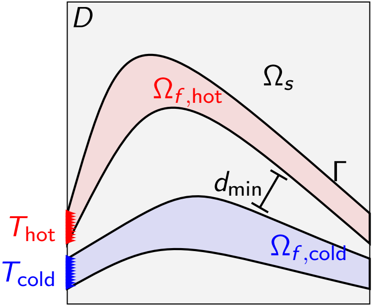
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



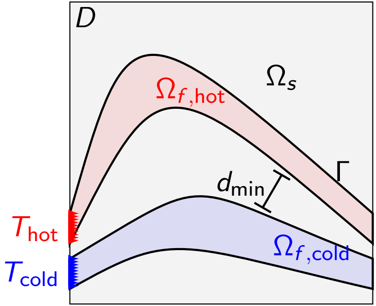
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



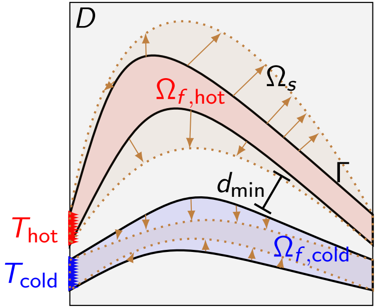
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



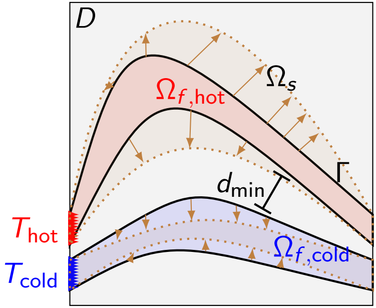
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



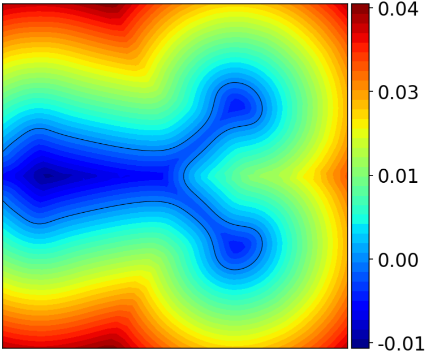
Outline

1. Formulation of the optimal heat exchanger design problem
    - Physical modelling
    - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
    - Shape derivatives
    - Treatment of geometric constraints
3. Numerical Topology optimization
    - Null space optimization algorithm
    - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results

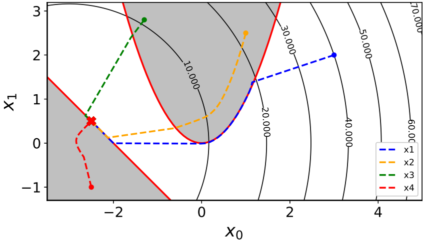


Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results

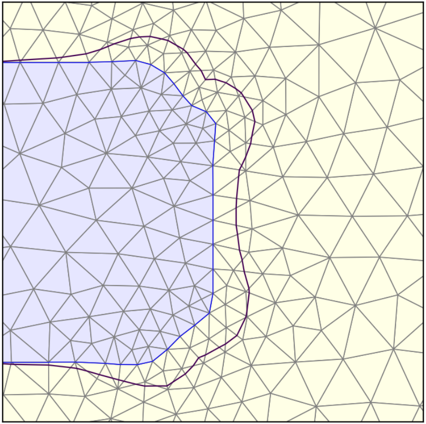
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



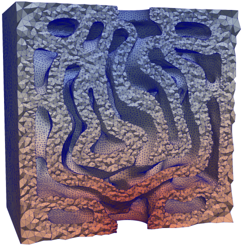
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



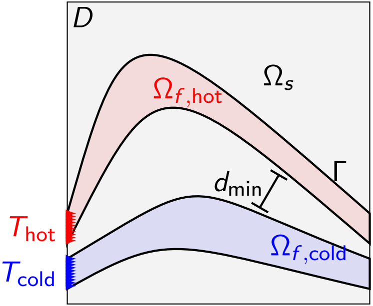
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling



Problem at hand

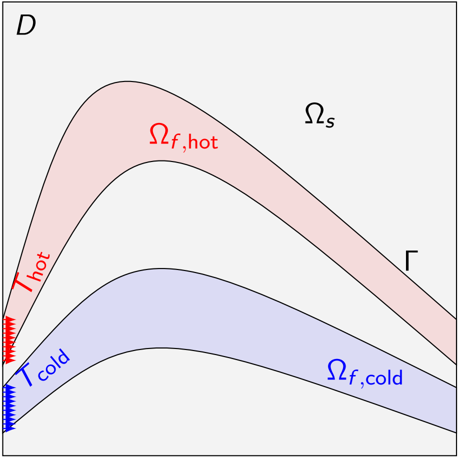


Figure: Settings of the heat exchanger topology optimization problem.

Problem at hand

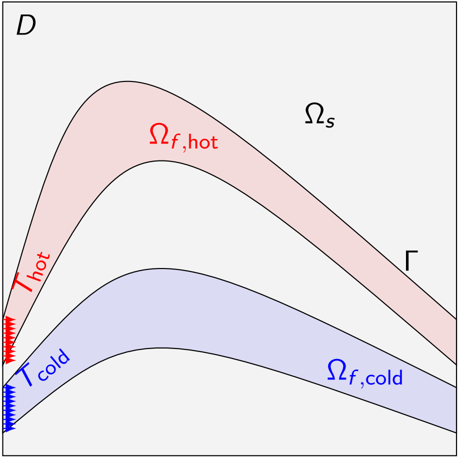


Figure: Settings of the heat exchanger topology optimization problem.

- Navier-Stokes flows in the hot and cold phases $\Omega_{f,hot}$ and $\Omega_{f,cold}$.

Problem at hand

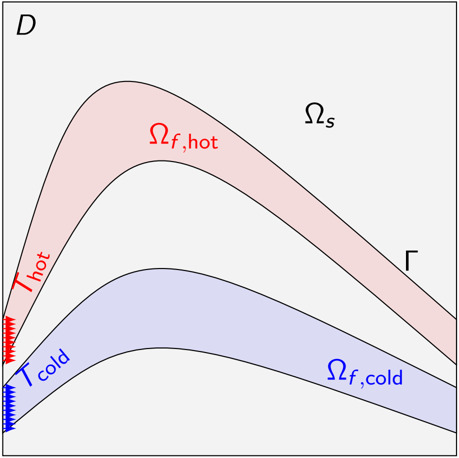


Figure: Settings of the heat exchanger topology optimization problem.

- Navier-Stokes flows in the hot and cold phases $\Omega_{f,hot}$ and $\Omega_{f,cold}$.
- Thermal convection in the fluid phase $\Omega_f = \Omega_{f,hot} \cup \Omega_{f,cold}$.
- Thermal diffusion in Ω_s and Ω_f with conductivities $k_s >> k_f$.

Problem at hand

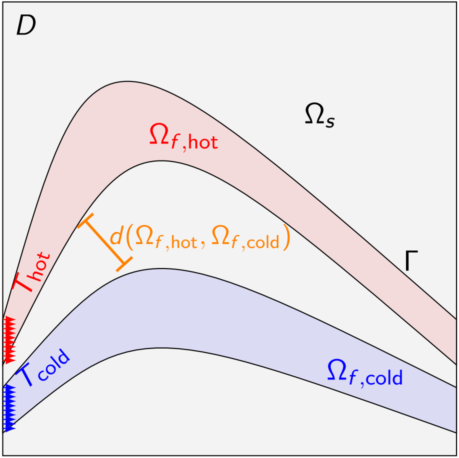


Figure: Settings of the heat exchanger topology optimization problem.

- Navier-Stokes flows in the hot and cold phases $\Omega_{f,hot}$ and $\Omega_{f,cold}$.
- Thermal convection in the fluid phase $\Omega_f = \Omega_{f,hot} \cup \Omega_{f,cold}$.
- Thermal diffusion in Ω_s and Ω_f with conductivities $k_s >> k_f$.
- Non-penetration constraint:

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

Problem at hand

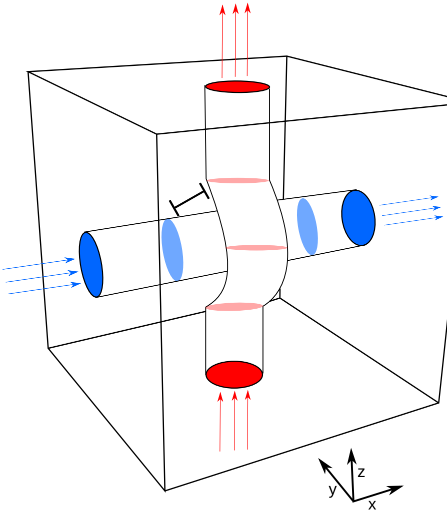


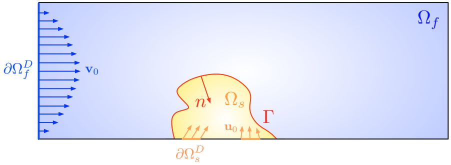
Figure: Settings of the heat exchanger topology optimization problem.

- Navier-Stokes flows in the hot and cold phases $\Omega_{f,hot}$ and $\Omega_{f,cold}$.
- Thermal convection in the fluid phase $\Omega_f = \Omega_{f,hot} \cup \Omega_{f,cold}$.
- Thermal diffusion in Ω_s and Ω_f with conductivities $k_s >> k_f$.
- Non-penetration constraint:

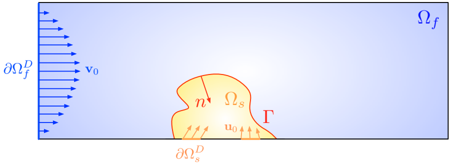
$$d(\Omega_{f,\text{hot}}, \Omega_{f,\text{cold}}) \geqslant d_{\text{min}}.$$

- In 3D!

The coupled physics model



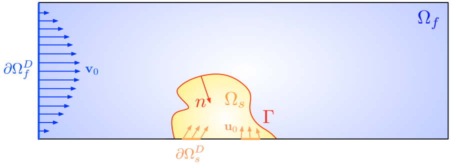
The coupled physics model



- Incompressible Navier-Stokes system for the velocity and pressure $(\boldsymbol{v}, p)$ in Ω_f

$$
\begin{aligned}
-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} &= \boldsymbol{f}_f \text{ in } \Omega_f \\
\mathrm{div}(\boldsymbol{v}) &= 0 \text{ in } \Omega_f
\end{aligned}
$$

The coupled physics model



- Incompressible Navier-Stokes system for the velocity and pressure $(\boldsymbol{v}, p)$ in Ω_f

$$-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = \boldsymbol{f}_f \text{ in } \Omega_f$$
$$\mathrm{div}(\boldsymbol{v}) = 0 \text{ in } \Omega_f$$

- Convection-diffusion for the temperature T in Ω_f and Ω_s:

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

Physical solvers

- The body-fitted approach uses discretization meshes with the fluid interface explicitly discritized.

Physical solvers

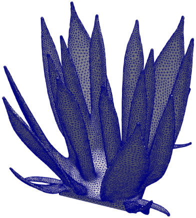
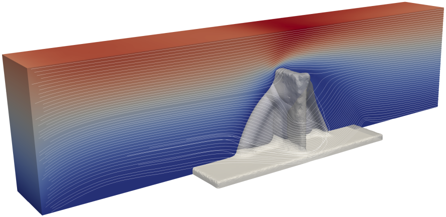
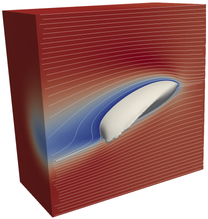
- The body-fitted approach uses discretization meshes with the fluid interface explicitly discritized. Allows to use any external solvers in principle.

Physical solvers

- The body-fitted approach uses discretization meshes with the fluid interface explicitly discritized. Allows to use any external solvers in principle.

For 3D applications, absolute need of parallel computing.

- We rely on finite element discretization of the weak formulation and a Newton scheme for Navier-Stokes problem
- We Use Domain Decomposition and adapted preconditioners for solving finite element problems : all FEM related operations are achieved in parallel with FreeFEM.

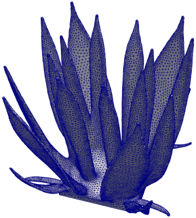
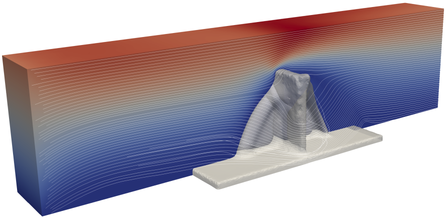
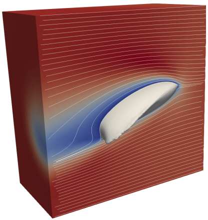




Physical solvers

- The body-fitted approach uses discretization meshes with the fluid interface explicitly discritized. Allows to use any external solvers in principle.

For 3D applications, absolute need of parallel computing.

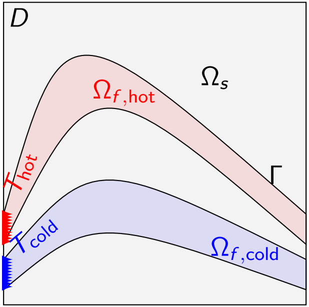
- We rely on finite element discretization of the weak formulation and a Newton scheme for Navier-Stokes problem
- We Use Domain Decomposition and adapted preconditioners for solving finite element problems : all FEM related operations are achieved in parallel with FreeFEM.
- Our examples feature fluid FEM problems on meshes up to 5 millions of Tetrahedra with 30 CPUs.

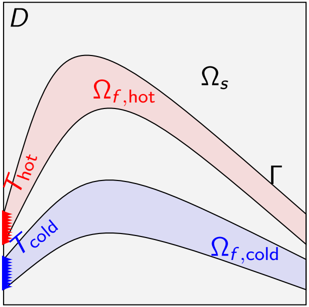
The optimal design problem

- Heat exchanged:

$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) := \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y,$$



The optimal design problem



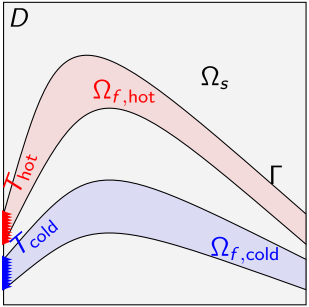
- Heat exchanged:

$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) := \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T\, \boldsymbol{v}\cdot\boldsymbol{n}\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T\, \boldsymbol{v}\cdot\boldsymbol{n}\mathrm{d}y,$$

- Pressure drop:

$$\mathtt{DP}(\Omega_{f,\text{cold}}, p(\Omega_f)) := \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{in}}} p\,\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{out}}} p\,\mathrm{d}y$$

The optimal design problem



- Heat exchanged:

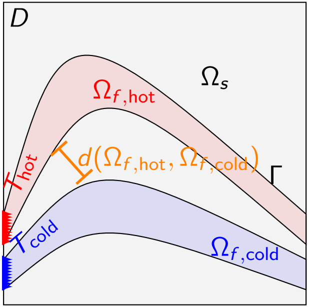
$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) := \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T\, \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T\, \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}y,$$

- Pressure drop:

$$\mathtt{DP}(\Omega_{f,\text{cold}}, p(\Omega_f)) := \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{in}}} p\,\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{out}}} p\,\mathrm{d}y$$

- Non mixing constraint

The optimal design problem



- Heat exchanged:

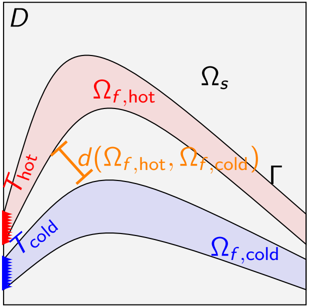
$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) := \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y,$$

- Pressure drop:

$$\mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) := \int_{\partial\Omega_{f,\mathrm{cold}} \cap \partial\Omega_{f,\mathrm{in}}} p \, \mathrm{d}y - \int_{\partial\Omega_{f,\mathrm{cold}} \cap \partial\Omega_{f,\mathrm{out}}} p \, \mathrm{d}y$$

- Non mixing constraint

The optimal design problem



- Heat exchanged:

$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) \\ := \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T \, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y,$$

- Pressure drop:

$$\mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \\ := \int_{\partial\Omega_{f,\mathrm{cold}} \cap \partial\Omega_{f,\mathrm{in}}} p \, \mathrm{d}y - \int_{\partial\Omega_{f,\mathrm{cold}} \cap \partial\Omega_{f,\mathrm{out}}} p \, \mathrm{d}y$$

- Non mixing constraint

$$d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) := \inf_{\substack{x \in \Omega_{f,\mathrm{cold}} \\ y \in \Omega_{f,\mathrm{hot}}}} |x - y| \geqslant d_{\min}.$$

The optimal design problem

The shape optimization problem:

$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \quad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\min}. \end{cases}$$

The optimal design problem

The shape optimization problem:

$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \qquad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

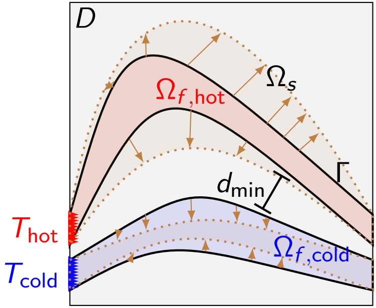
$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}}. \end{cases}$$

Optionally, mass constraints on fluids (or on solid):

$$\mathtt{Vol}(\Omega_{f,\mathrm{cold}}) \leq \mathtt{Vol}_0, \quad \mathtt{Vol}(\Omega_{f,\mathrm{hot}}) \leq \mathtt{Vol}_0.$$

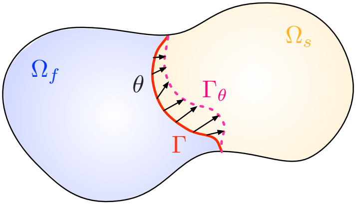
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard



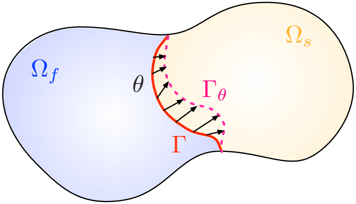
The boundary variation method of Hadamard

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



The boundary variation method of Hadamard

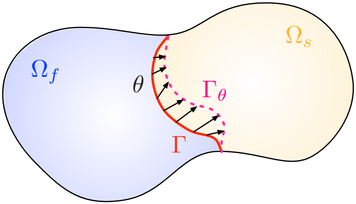
$$\min_{\Gamma} \; J(\Gamma)$$



$$\Gamma_{\boldsymbol{\theta}} = (I + \boldsymbol{\theta})\Gamma, \text{ with } \boldsymbol{\theta} \in W_0^{1,\infty}(D, \mathbb{R}^d), \; ||\boldsymbol{\theta}||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1.$$

The boundary variation method of Hadamard

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

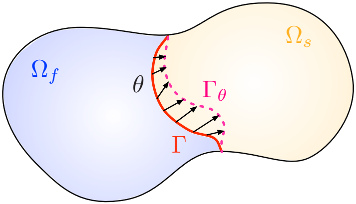


$$\Gamma_{\boldsymbol{\theta}} = (I + \boldsymbol{\theta})\Gamma, \text{ with } \boldsymbol{\theta} \in W_0^{1,\infty}(D, \mathbb{R}^d), \; ||\boldsymbol{\theta}||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1.$$

$$J(\Gamma_{\boldsymbol{\theta}}) = J(\Gamma) + \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}}(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \qquad \text{with } \frac{|o(\boldsymbol{\theta})|}{||\boldsymbol{\theta}||_{W^{1,\infty}(D,\mathbb{R}^d)}} \xrightarrow{\boldsymbol{\theta}\to 0} 0.$$

The boundary variation method of Hadamard

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



$$\Gamma_{\boldsymbol{\theta}} = (I + \boldsymbol{\theta})\Gamma, \text{ with } \boldsymbol{\theta} \in W_0^{1,\infty}(D, \mathbb{R}^d), \; ||\boldsymbol{\theta}||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1.$$

$$J(\Gamma_{\boldsymbol{\theta}}) = J(\Gamma) + \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}}(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \quad \text{with } \frac{|o(\boldsymbol{\theta})|}{||\boldsymbol{\theta}||_{W^{1,\infty}(D,\mathbb{R}^d)}} \xrightarrow{\boldsymbol{\theta}\to 0} 0.$$

Under suitable regularity assumptions, Hadamard structure theorem holds:

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}}(\Omega)(\boldsymbol{\theta}) = \int_{\partial\Omega} v_J(\Omega)\, \boldsymbol{\theta} \cdot \boldsymbol{n} \mathrm{d}y$$

for some $v_J(\Omega) \in L^1(\partial\Omega)$.

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]:

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]:

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

This is the volume expression of the shape derivative. There exists also a surface expression of the form $DF(\boldsymbol{\theta}) = \int_\Gamma v_F \boldsymbol{\theta} \cdot \boldsymbol{n} \mathrm{d}y$.

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]:

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

Two adjoint terms corresponding either of the two physics

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]:

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

Two adjoint terms corresponding either of the two physics

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]*:*

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

Depends on adjoint states S and $(\boldsymbol{w}, q)$ solved in inverse cascade.

Shape derivatives

Proposition

Let $F(\Omega_f, T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ an arbitrary cost function. If F has continuous partial derivatives, then $\Omega_f \mapsto F(\Omega_f, \boldsymbol{u}(\Omega_f), T(\Omega_f), \boldsymbol{v}(\Omega_f), p(\Omega_f))$ is shape differentiable and the shape derivative reads[2]*:*

$$
\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}\, \boldsymbol{v}) \mathrm{div}(\boldsymbol{\theta}) \mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v}, p) : (\nabla \boldsymbol{w} \nabla \boldsymbol{\theta}) + \sigma_f(\boldsymbol{w}, q) : (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta}) + \rho \boldsymbol{w} \cdot (\nabla \boldsymbol{v} \nabla \boldsymbol{\theta})\, \boldsymbol{v}] \mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T \cdot \nabla S) \mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T \cdot \nabla S + \rho c_p (\boldsymbol{v} \cdot \nabla T) S) \mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S \mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f (\nabla \boldsymbol{\theta} + \nabla \boldsymbol{\theta}^T) \nabla T \cdot \nabla S + \rho c_p \boldsymbol{v} \cdot (\nabla \boldsymbol{\theta}^T \nabla T) S \right] \mathrm{d}x.
\end{aligned}
$$

Adjoint system

Adjoint states in variational formulation fed with partial derivatives :

$$\int_{\Omega_s} k_s \nabla S \cdot \nabla S' \mathrm{d}x + \int_{\Omega_f} (k_f \nabla S \cdot \nabla S' + \rho c_p S \boldsymbol{v} \cdot \nabla S') \mathrm{d}x = \frac{\partial F}{\partial T}(S).$$

Adjoint system

Adjoint states in variational formulation fed with partial derivatives :

$$\int_{\Omega_s} k_s \nabla S \cdot \nabla S' \mathrm{d}x + \int_{\Omega_f} (k_f \nabla S \cdot \nabla S' + \rho c_p S \boldsymbol{v} \cdot \nabla S') \mathrm{d}x = \frac{\partial F}{\partial T}(S).$$

$$\forall (\boldsymbol{w}', q') \in V_{\boldsymbol{v},p}$$

$$\int_{\Omega_f} \Big(\sigma_f(\boldsymbol{w}, q) : \nabla \boldsymbol{w}' + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{w}' \cdot \boldsymbol{v} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v} \cdot \boldsymbol{w}' - q' \mathrm{div}(\boldsymbol{w}) \Big) \mathrm{d}x =$$

$$\int_{\Omega_f} -\rho c_p S \nabla T \cdot \boldsymbol{w}' \mathrm{d}x + \frac{\partial F}{\partial (\boldsymbol{v}, p)}(\boldsymbol{w}', q'),$$

Adjoint system

Adjoint states in variational formulation fed with partial derivatives :

$$\int_{\Omega_s} k_s \nabla S \cdot \nabla S' \mathrm{d}x + \int_{\Omega_f} (k_f \nabla S \cdot \nabla S' + \rho c_p S \boldsymbol{v} \cdot \nabla S') \mathrm{d}x = \frac{\partial F}{\partial T}(S).$$

$$\forall (\boldsymbol{w}', q') \in V_{\boldsymbol{v},p}$$

$$\int_{\Omega_f} \Big(\sigma_f(\boldsymbol{w}, q) : \nabla \boldsymbol{w}' + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{w}' \cdot \boldsymbol{v} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v} \cdot \boldsymbol{w}' - q' \mathrm{div}(\boldsymbol{w}) \Big) \mathrm{d}x =$$

$$\int_{\Omega_f} -\rho c_p S \nabla T \cdot \boldsymbol{w}' \mathrm{d}x + \frac{\partial F}{\partial(\boldsymbol{v}, p)}(\boldsymbol{w}', q'),$$

This can be implemented once for all and allows for easy changes of objective functions.

Adjoint system

This allows to compute the shape derivatives of the heat transfer and of the pressure drop.

$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \quad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}}. \end{cases}$$

Adjoint system

This allows to compute the shape derivatives of the heat transfer and of the pressure drop.

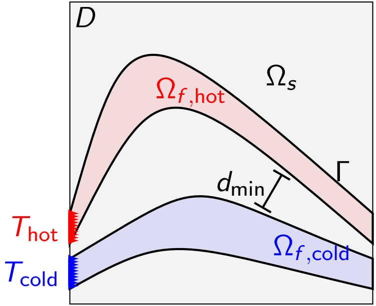
$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \quad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\min}. \end{cases}$$

Geometric constraints need a special treatment.

Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint



Outline

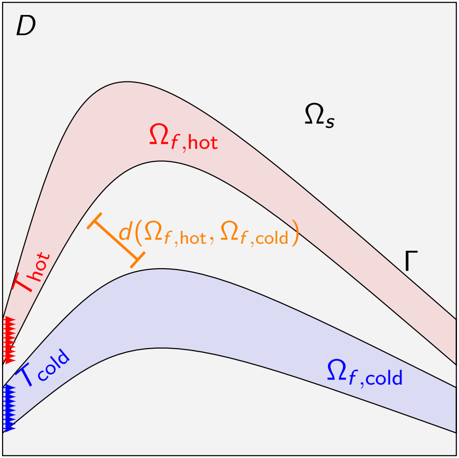


Figure: Settings of the heat exchanger topology optimization problem .

Non-penetration constraint:

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

Outline

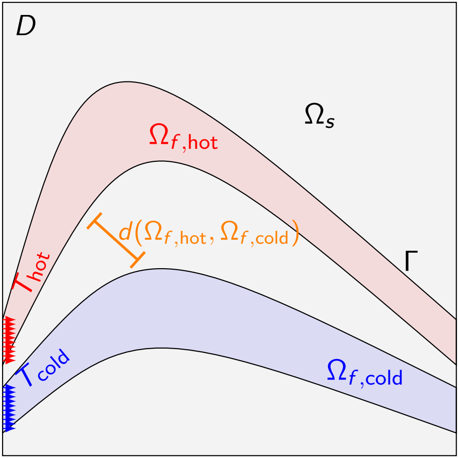


Figure: Settings of the heat exchanger topology optimization problem .

Non-penetration constraint:

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

We enforce it by imposing

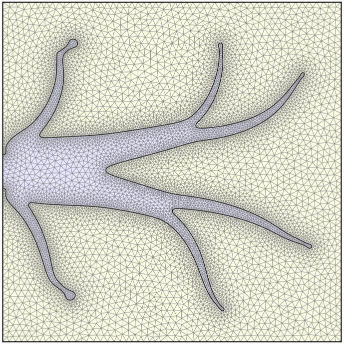
$$\forall x \in \Omega_{f,\mathrm{cold}},\ d_{\Omega_{f,\mathrm{hot}}}(x) \geqslant d_{\min},$$

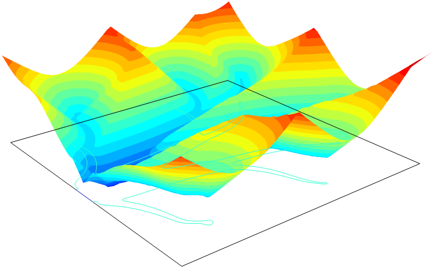
where $d_{\Omega_{f,\mathrm{hot}}}$ is the signed distance function to the domain $\Omega_{f,\mathrm{hot}}$.

The signed distance function

The signed distance function d_Ω to the domain $\Omega \subset D$ is defined by:

$$\forall x \in D, \;\; d_\Omega(x) = \begin{cases} -\min\limits_{y \in \partial\Omega} ||y - x|| & \text{if } x \in \Omega, \\ \min\limits_{y \in \partial\Omega} ||y - x|| & \text{if } x \in D \backslash \Omega. \end{cases}$$





The signed distance function

This allows to compute the shape derivatives of the heat transfer and of the pressure drop.

$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \qquad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}}. \end{cases}$$

Geometric constraints need a special treatment.

Non mixing constraint

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

Non mixing constraint

$$d(\Omega_{f,\text{cold}}, \Omega_{f,\text{hot}}) \geqslant d_{\mathsf{min}} \Leftrightarrow \forall y \in \partial\Omega_{f,\text{cold}},\ d_{\Omega_{f,\text{hot}}}(y) \geqslant d_{\mathsf{min}}$$

Non mixing constraint

$$d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}} \Leftrightarrow \forall y \in \partial\Omega_{f,\mathrm{cold}},\ d_{\Omega_{f,\mathrm{hot}}}(y) \geqslant d_{\mathsf{min}}$$

This is equivalent to

$$\left\| \frac{1}{d_{\Omega_{f,\mathrm{hot}}}} \right\|_{L^\infty(\partial\Omega_{f,\mathrm{cold}})}^{-1} \geqslant d_{\mathsf{min}}$$

Non mixing constraint

$$d(\Omega_{f,\mathrm{cold}},\Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}} \Leftrightarrow \forall y \in \partial\Omega_{f,\mathrm{cold}},\ d_{\Omega_{f,\mathrm{hot}}}(y) \geqslant d_{\mathsf{min}}$$

This is equivalent to

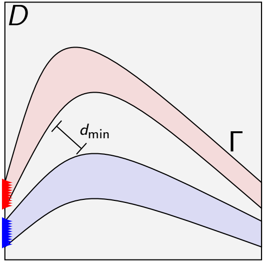
$$\left\| \frac{1}{d_{\Omega_{f,\mathrm{hot}}}} \right\|^{-1}_{L^\infty(\partial\Omega_{f,\mathrm{cold}})} \geqslant d_{\mathsf{min}}$$

We approximate the infinity norm with an averaged p-norm:

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

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

2D heat exchangers

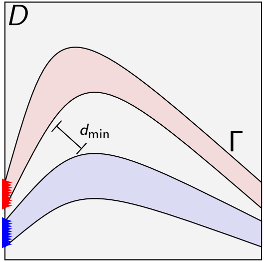


Heat exchanger problem with limited pressure loss and non-mixing constraint:

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

$$s.c. \begin{cases} \mathtt{DP}(\Omega_f) = \int_{\partial\Omega_f^D} p \mathrm{d}s - \int_{\partial\Omega_f^N} p \mathrm{d}s \leq \mathtt{DP}_0 \\ \qquad\qquad P_{\text{cold}\to\text{hot}}(\Omega_f) \geqslant d_{\min}. \end{cases}$$

2D heat exchangers

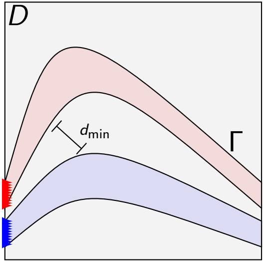


Heat exchanger problem with limited pressure loss and non-mixing constraint:

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

$$s.c. \begin{cases} \mathtt{DP}(\Omega_f) = \int_{\partial\Omega_f^D} p \mathrm{d}s - \int_{\partial\Omega_f^N} p \mathrm{d}s \leq \mathtt{DP}_0 \\ P_{\text{cold}\to\text{hot}}(\Omega_f) = \int_D j(d_{\Omega_{f,hot}}) \mathrm{d}x \geqslant d_{\min}. \end{cases}$$

2D heat exchangers



Heat exchanger problem with limited pressure loss and non-mixing constraint:

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

$$s.c. \begin{cases} \mathrm{DP}(\Omega_f) = \int_{\partial\Omega_f^D} p \mathrm{d}s - \int_{\partial\Omega_f^N} p \mathrm{d}s \leq \mathrm{DP}_0 \\ P_{\text{cold}\to\text{hot}}(\Omega_f) = \int_D j(d_{\Omega_{f,hot}}) \mathrm{d}x \geqslant d_{\min}. \end{cases}$$

What is the shape derivative of $P_{\text{cold}\to\text{hot}}(\Omega_f)$?

Shape derivatives of geometric constraints

This reduces to the setting of computing the shape derivative of $d_{\Omega_{f,\mathrm{hot}}}$, $P_{hot\leftrightarrow cold}(\Omega_f)$ with:

$$P_{hot\leftrightarrow cold}(\Omega_f) := \int_D j(d_{\Omega_{f,hot}})\mathrm{d}x.$$

Shape derivatives of geometric constraints

The shape derivative of $P_{hot \leftrightarrow cold}(\Omega_f)$ is given by[3]:

$$P'_{hot \leftrightarrow cold}(\Omega)(\boldsymbol{\theta}) = \int_{\partial \Omega_{f,hot}} u(y)\, \boldsymbol{\theta} \cdot \boldsymbol{n} \, \mathrm{d}y$$

$$\text{with } u(y) = -\int_{z \in \mathrm{ray}(y)} j'(d_{\Omega_{f,hot}}(z)) \prod_{1 \le i \le n-1} (1 + \kappa_i(y) d_{\Omega_{f,hot}}(z)) \mathrm{d}z, \qquad \forall y \in \partial\Omega.$$

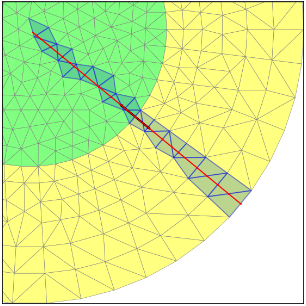
[3] Allaire, Jouve, and Michailidis, *Thickness control in structural optimization via a level set method* (2016)

Shape derivatives of geometric constraints

The shape derivative of $P_{hot\leftrightarrow cold}(\Omega_f)$ is given by[3]:

$$P'_{hot\leftrightarrow cold}(\Omega)(\boldsymbol{\theta}) = \int_{\partial\Omega_{f,hot}} u(y)\, \boldsymbol{\theta}\cdot\boldsymbol{n}\,\mathrm{d}y$$

$$\text{with } u(y) = -\int_{z\in\mathrm{ray}(y)} j'(d_{\Omega_{f,hot}}(z)) \prod_{1\le i\le n-1} (1+\kappa_i(y) d_{\Omega_{f,hot}}(z))\mathrm{d}z, \qquad \forall y\in\partial\Omega.$$



The computation of $u(y)$ requires a priori integration along the normal rays and the computation of curvatures/u/ffeppon$\kappa_i(y)$.

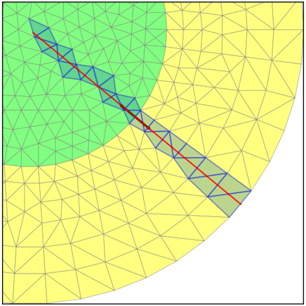
[3] Allaire, Jouve, and Michailidis, *Thickness control in structural optimization via a level set method* (2016)

Shape derivatives of geometric constraints

The shape derivative of $P_{hot\leftrightarrow cold}(\Omega_f)$ is given by[3]:

$$P'_{hot\leftrightarrow cold}(\Omega)(\boldsymbol{\theta}) = \int_{\partial\Omega_{f,hot}} u(y)\, \boldsymbol{\theta}\cdot\boldsymbol{n}\,\mathrm{d}y$$

$$\text{with } u(y) = -\int_{z\in\mathrm{ray}(y)} j'(d_{\Omega_{f,hot}}(z)) \prod_{1\le i\le n-1}(1+\kappa_i(y)d_{\Omega_{f,hot}}(z))\mathrm{d}z, \qquad \forall y\in\partial\Omega.$$



The computation of $u(y)$ requires a priori integration along the normal rays and the computation of curvatures/u/ffeppon$\kappa_i(y)$.

[3] Allaire, Jouve, and Michailidis, *Thickness control in structural optimization via a level set method* (2016)

Shape derivatives of geometric constraints

It turns out that it is possible to compute u without integrating along the rays[4]:

[4] Feppon, Allaire, and Dapogny, *A variational formulation for computing shape derivatives of geometric constraints along rays* (2019)

Shape derivatives of geometric constraints

It turns out that it is possible to compute u without integrating along the rays[4]:

Proposition

Let $\hat{u} \in V_\omega$ *be the solution to the variational problem*

$$\forall v \in V_\omega,\ \int_{\partial\Omega_{f,hot}} \hat{u} v \mathrm{d}s + \int_D \omega(\nabla d_{\Omega_{f,hot}} \cdot \nabla \hat{u})(\nabla d_{\Omega_{f,hot}} \cdot \nabla v)\mathrm{d}x = -\int_D j'(d_{\Omega_{f,hot}}) v \mathrm{d}x.$$

Then $u(y) = \hat{u}(y)$ *for any* $y \in \partial\Omega_{f,hot}$.

[4]Feppon, Allaire, and Dapogny, *A variational formulation for computing shape derivatives of geometric constraints along rays* (2019)

Shape derivatives of geometric constraints

It turns out that it is possible to compute u without integrating along the rays[4]:

Proposition

Let $\hat{u} \in V_\omega$ *be the solution to the variational problem*

$$\forall v \in V_\omega,\ \int_{\partial\Omega_{f,hot}} \hat{u} v \mathrm{d}s + \int_D \omega(\nabla d_{\Omega_{f,hot}} \cdot \nabla \hat{u})(\nabla d_{\Omega_{f,hot}} \cdot \nabla v)\mathrm{d}x = -\int_D j'(d_{\Omega_{f,hot}}) v \mathrm{d}x.$$

Then $u(y) = \hat{u}(y)$ *for any* $y \in \partial\Omega_{f,hot}$.

- This variational problem can easily be solved with FEM in 2-D and 3D

[4] Feppon, Allaire, and Dapogny, *A variational formulation for computing shape derivatives of geometric constraints along rays* (2019)

Shape derivatives of geometric constraints

It turns out that it is possible to compute u without integrating along the rays[4]:

Proposition

Let $\hat{u} \in V_\omega$ *be the solution to the variational problem*

$$\forall v \in V_\omega, \int_{\partial\Omega_{f,hot}} \hat{u} v \mathrm{d}s + \int_D \omega(\nabla d_{\Omega_{f,hot}} \cdot \nabla \hat{u})(\nabla d_{\Omega_{f,hot}} \cdot \nabla v)\mathrm{d}x = -\int_D j'(d_{\Omega_{f,hot}}) v \mathrm{d}x.$$

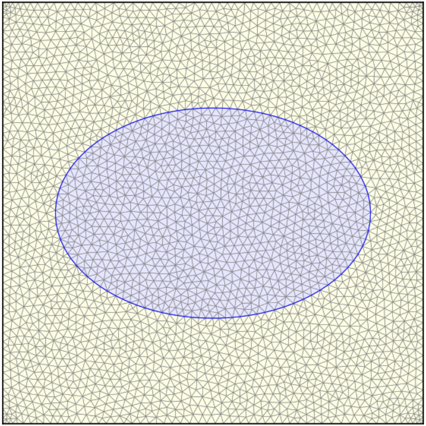
Then $u(y) = \hat{u}(y)$ *for any* $y \in \partial\Omega_{f,hot}$.

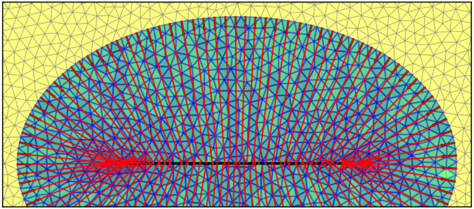
- This variational problem can easily be solved with FEM in 2-D and 3D
- This allows to handle conveniently geometric constraints (e.g. maximum thickness, minum distance, etc. . .) in 2D and 3D level set based topology optimization.

[4] Feppon, Allaire, and Dapogny, *A variational formulation for computing shape derivatives of geometric constraints along rays* (2019)

Shape derivatives of geometric constraints

A comparison with an analytic case:





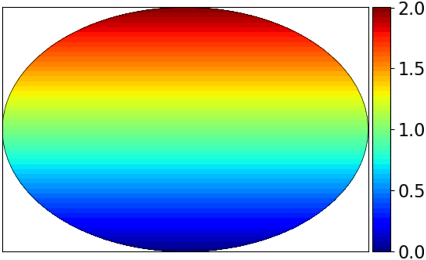
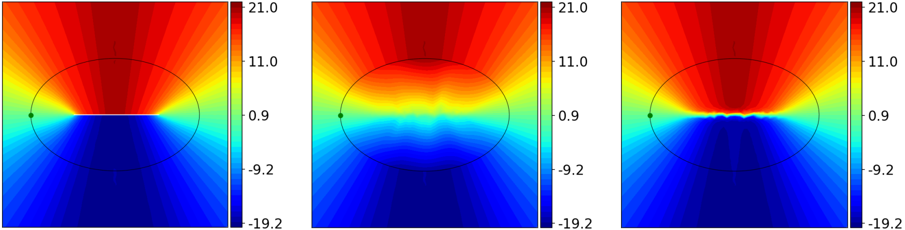


Figure: A prescribed $-j'(d_\Omega(x))$.

Shape derivatives of geometric constraints

The weight ω needs to vanish near the skeleton (medial axis).

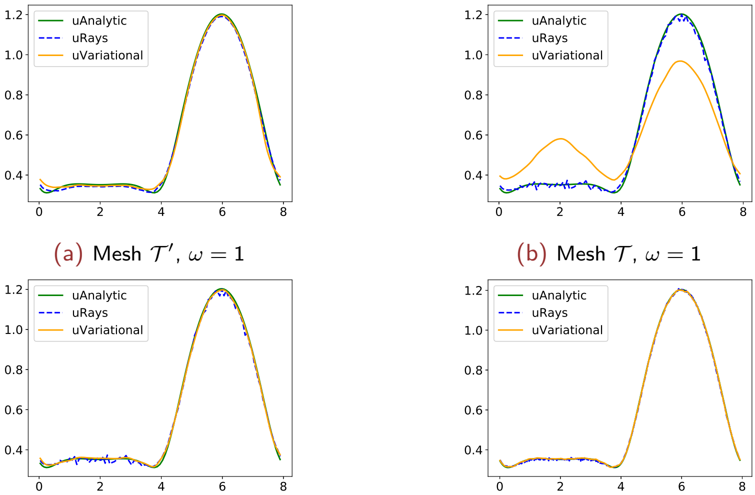


(a) Mesh $\mathcal{T}'$ (manually removed skeleton), $\omega = 1$

(b) Mesh $\mathcal{T}$, $\omega = 1$.

(c) Mesh $\mathcal{T}$, $\omega = 2/(1 + |\Delta d_\Omega|^{3.5})$

Shape derivatives of geometric constraints



(a) Mesh $\mathcal{T}'$, $\omega = 1$

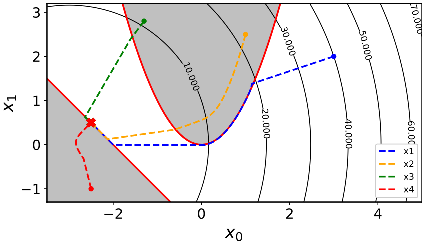
(b) Mesh $\mathcal{T}$, $\omega = 1$

(c) Mesh $\mathcal{T}$, $\omega = 2/(1 + |\Delta d_\Omega|^{3.5})$

(d) Finer mesh $\mathcal{T}$, $\omega = 2/(1 + |\Delta d_\Omega|^{3.5})$

Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm



Null space gradient flows

- Our goal: solve constrained shape optimization problems

$$\begin{aligned} \min_{\Gamma} \quad & J(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) \\ \text{s.t.} \quad & g_i(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) = 0, \ 1 \leq i \leq p \cdot \\ & h_i(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) \leq 0, \ 1 \leq i \leq q \end{aligned}$$

with *arbitrary* functionals J, g_i, h_i;

Null space gradient flows

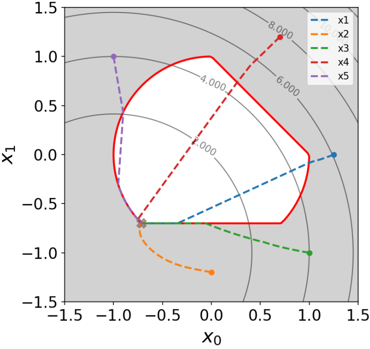
- Our goal: solve constrained shape optimization problems

$$\begin{aligned} \min_{\Gamma} \quad & J(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) \\ \text{s.t.} \quad & g_i(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) = 0, \, 1 \leq i \leq p \cdot \\ & h_i(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma)) \leq 0, \, 1 \leq i \leq q \end{aligned}$$

  with *arbitrary* functionals J, g_i, h_i;
- g_i and h_i represent industrial specification constraints (mass, pressure drop...)

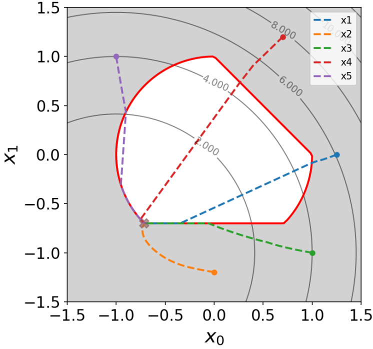
Null space optimization algorithm

- Nonlinear constrained optimization on manifolds with a moderate number of constraints



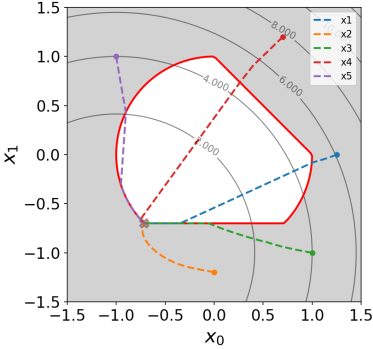
Null space optimization algorithm

- Nonlinear constrained optimization on manifolds with a moderate number of constraints
- Generalization of the unconstrained gradient flow: no hard tuning of parameters



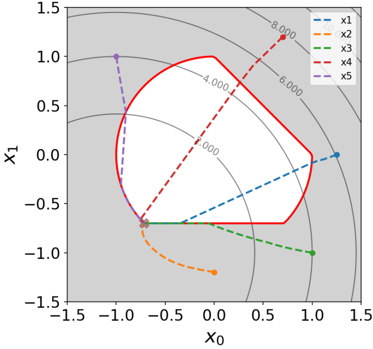
Null space optimization algorithm

- Nonlinear constrained optimization on manifolds with a moderate number of constraints
- Generalization of the unconstrained gradient flow: no hard tuning of parameters
- Gradual corrections of unfeasible initializations



Null space optimization algorithm

- Nonlinear constrained optimization on manifolds with a moderate number of constraints
- Generalization of the unconstrained gradient flow: no hard tuning of parameters
- Gradual corrections of unfeasible initializations
- Adapted to the infinite dimensional setting of the method of Hadamard



Null space optimization algorithm

For the exposure of our method, let us consider

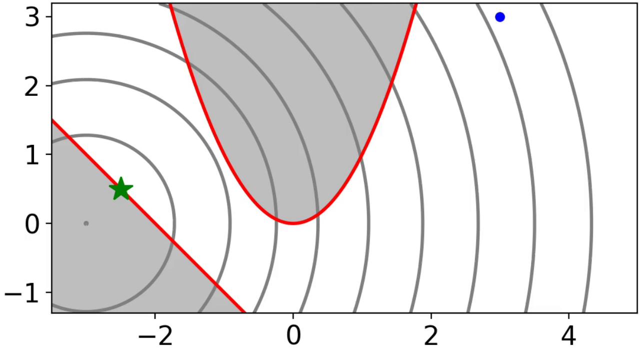
$$\min_{x\in\mathbb{R}^n} \quad J(x)$$
$$\text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases}$$

with $J : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^q$ Fréchet differentiable.

Null space optimization algorithm

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

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



Null space optimization algorithm

We extend classical dynamical systems approaches to constrained optimization:

Null space optimization algorithm

We extend classical dynamical systems approaches to constrained optimization:

- For unconstrained optimization, the celebrated gradient flow:

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

Null space optimization algorithm

We extend classical dynamical systems approaches to constrained optimization:

- For unconstrained optimization, the celebrated gradient flow:

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

- 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 $V = \{x \in V \,|\, \boldsymbol{g}(x) = 0\}$)

Null space optimization algorithm

We extend classical dynamical systems approaches to constrained optimization:

- For unconstrained optimization, the celebrated gradient flow:

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

- 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 $V = \{x \in V \mid \boldsymbol{g}(x) = 0\}$) Then Yamashita (1980) added a Gauss-Newton direction:

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

Null space optimization algorithm

We extend classical dynamical systems approaches to constrained optimization:

- For unconstrained optimization, the celebrated gradient flow:

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

- 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 $V = \{x \in V \,|\, \boldsymbol{g}(x) = 0\}$) 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 $J(x(t))$ decreases if $\boldsymbol{g}(x(t)) = 0$.

Null space optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, we consider:

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

with

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

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

Null space optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, we consider:

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

with

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

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

$\widetilde{I}(x)$ the set of violated constraints:

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

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

Null space optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, we consider:

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

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 an "optimal" subset of the active or violated constraints which can be computed by mean of a dual subproblem.

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

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

Null space optimization algorithm

For *both* equality constraints $\boldsymbol{g}(x) = 0$ and inequality constraints $\boldsymbol{h}(x) \leq 0$, we consider:

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

with

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

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

The transpose $\cdot^T$ operator encompasses the regularization and extension step of the shape derivative.

Null space optimization algorithm

We can prove:

1. Constraints are asymptotically satisfied:

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

Null space optimization algorithm

We can prove:

1. Constraints are asymptotically satisfied:

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

Null space 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

Null space 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

Null space 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

Null space 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

Null space optimization algorithm

We can prove:

1. Constraints are asymptotically satisfied:

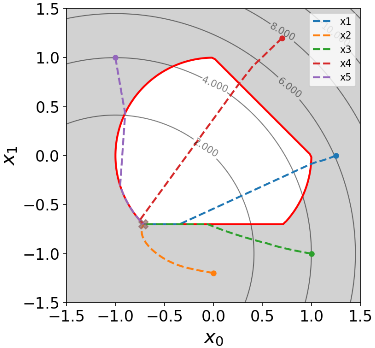
$$\boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x(0)) \text{ and } \boldsymbol{h}_{\widetilde{I}(x(t))} \leq e^{-\alpha_C t}\boldsymbol{h}(x(0))$$

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

The algorithm can be adapted to the infinite-dimensional shape optimization framework.

Null space optimization algorithm

Try it yourself! Open source implementation[5]:



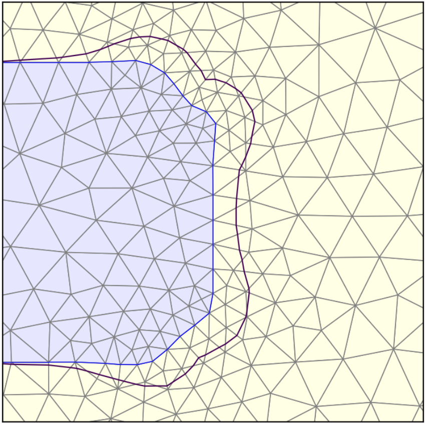
`https://gitlab.com/florian.feppon/null-space-optimizer`

`pip install nullspace_optimizer`

[5]Feppon, Allaire, and Dapogny, *Null space gradient flows for constrained optimization with applications to shape optimization* (2019)

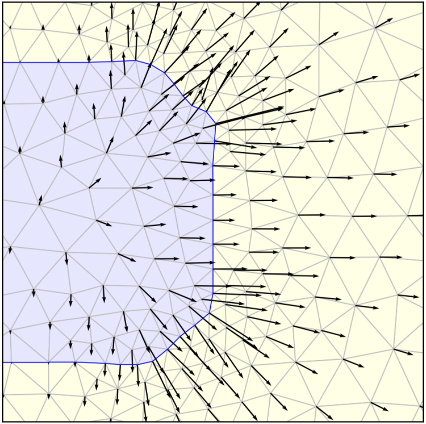
Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates



Body-fitted meshes

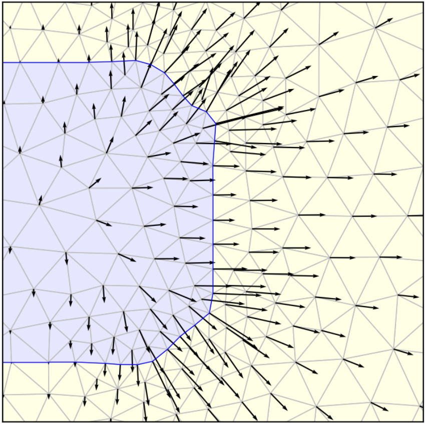
We rely on body fitted meshes[6,7].



Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.

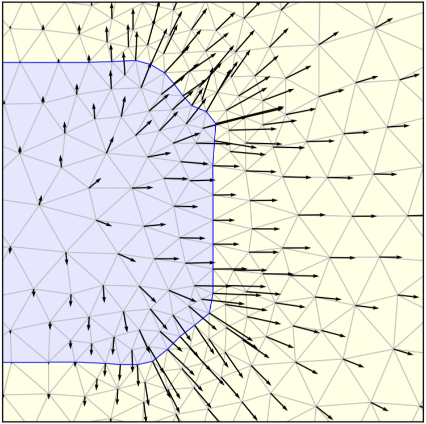


6

Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control.



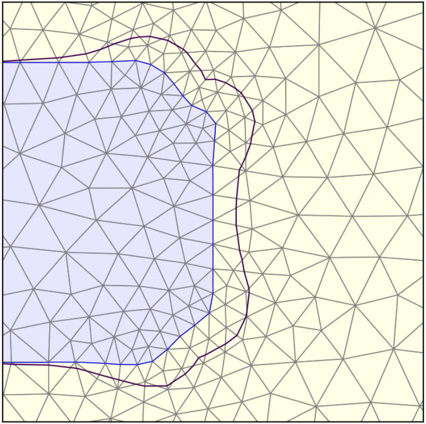
Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control. Solve the level-set equation

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

$$\Omega_f(t) = \{x \in D \,|\, \phi(t,x) \leq 0\}.$$



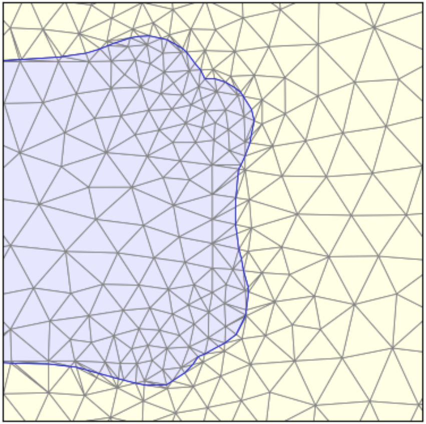
Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control. Solve the level-set equation

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

$$\Omega_f(t) = \{x \in D \,|\, \phi(t,x) \leq 0\}.$$



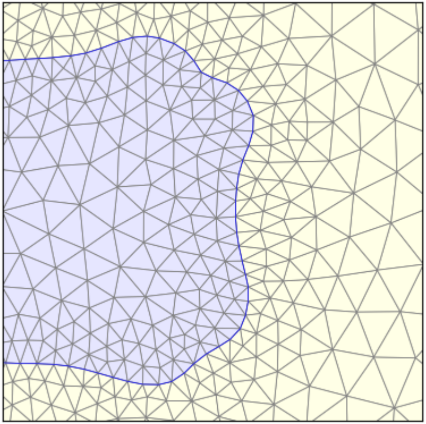
Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control. Solve the level-set equation

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

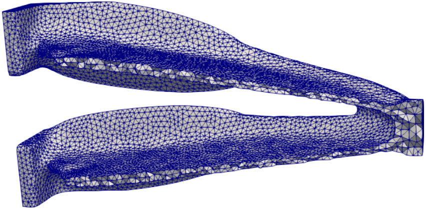
$$\Omega_f(t) = \{x \in D \mid \phi(t,x) \leq 0\}.$$



Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control.
- Allows to capture very fine details.



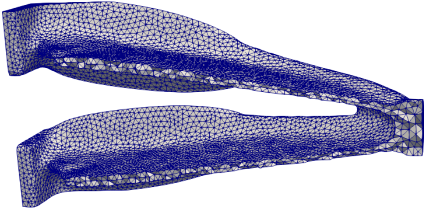
[6] Allaire, Dapogny, and Frey, *Shape optimization with a level set based mesh evolution method* (2014)

[7] Feppon et al., *Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework* (2019)

Body-fitted meshes

We rely on body fitted meshes[6,7].

- Fluid-Solid interface Γ exactly captured, no need of physics interpolation because no porous regions.
- Remeshing with `Mmg` enabling mesh size control.
- Allows to capture very fine details.



Remark: Mesh adaptation and Isosurface discretization in Mmg is still sequential. A future release of (Par)Mmg will allow to remesh in parallel.

[6] Allaire, Dapogny, and Frey, *Shape optimization with a level set based mesh evolution method* (2014)

[7] Feppon et al., *Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework* (2019)

3D thermal diffusion

Maximization of heat conduction:

$$\min_{\Omega_f \subset D} \int_D T \mathrm{d}x$$
$$s.c. \quad \int_{\Omega_f} \mathrm{d}x \leq V_0$$

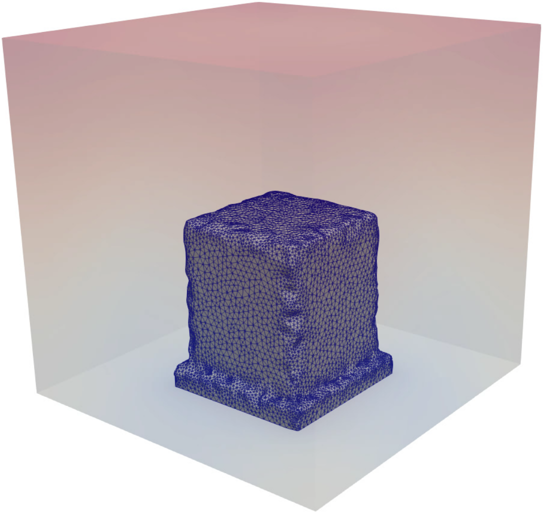
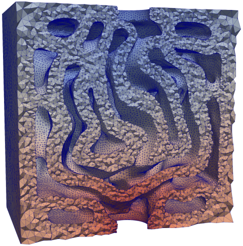


Figure: Thermal diffusion

Outline

1. Formulation of the optimal heat exchanger design problem
   - Physical modelling
   - Non-mixing constraint
2. Shape and Topology optimization with the method of Hadamard
   - Shape derivatives
   - Treatment of geometric constraints
3. Numerical Topology optimization
   - Null space optimization algorithm
   - Body fitted mesh evolution method for numerical shape updates
4. Numerical Results



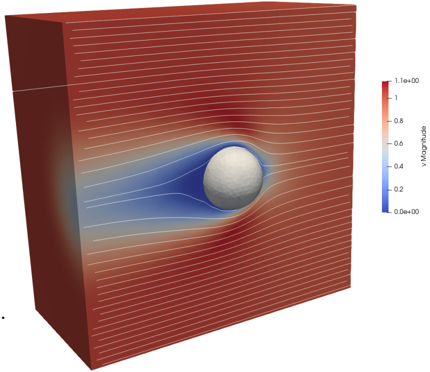
Lift-Drag optimization

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

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

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

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

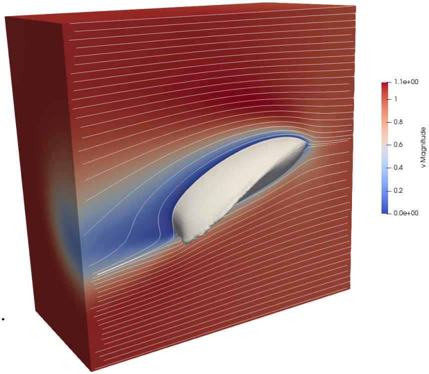


Lift-Drag optimization

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

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

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



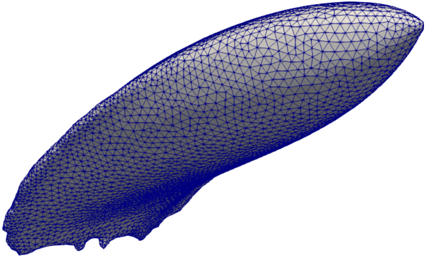
Lift-Drag optimization

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

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

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

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



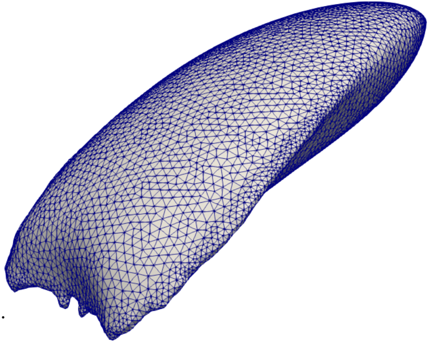
Lift-Drag optimization

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

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

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

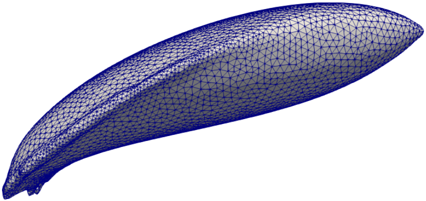
$$\operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v} \mathrm{d}x.$$



Lift-Drag optimization

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

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



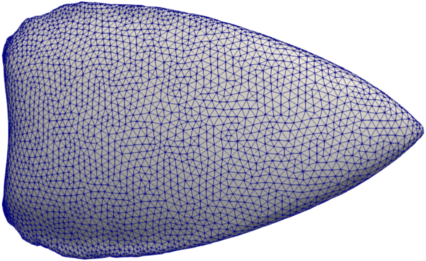
$$\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

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

Lift-Drag optimization

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

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

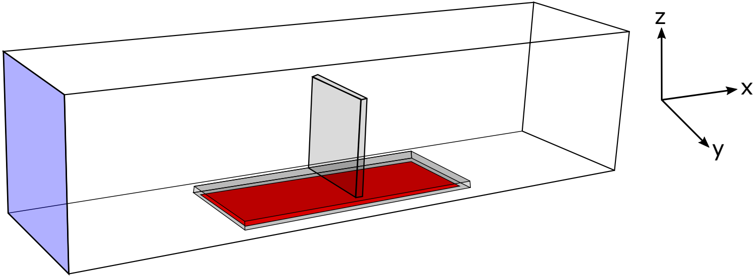


$$\mathrm{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \boldsymbol{n} \mathrm{d}s,$$

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

Fluid-structure interaction

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

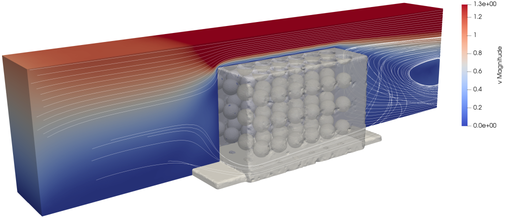


$$\min \quad \int_{\Omega_s} Ae(\boldsymbol{u}) : e(\boldsymbol{u}) \mathrm{d}x$$

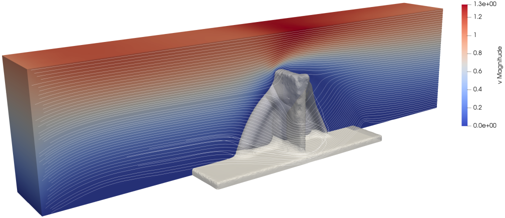
$$s.c. \quad \mathrm{Vol}(\Omega_s) = \mathrm{Vol}_{target}.$$

Fluid-structure interaction

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



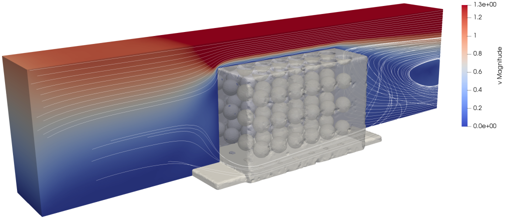
(a) Initial shape



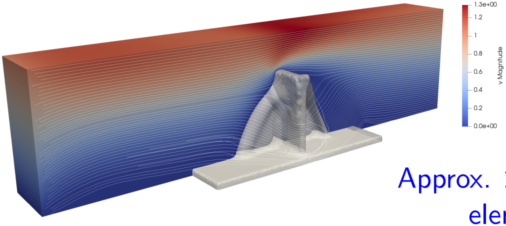
(b) Optimized shape

Fluid-structure interaction

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



(a) Initial shape



(b) Optimized shape

Approx. 2 millions of elements.

Fluid-structure interaction

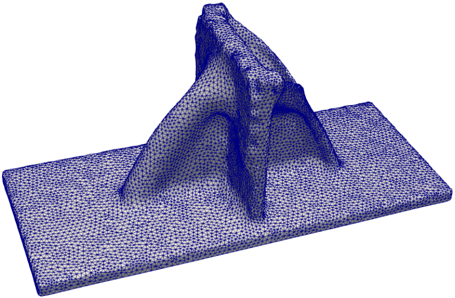


Figure: Optimized shape.

Fluid-structure interaction

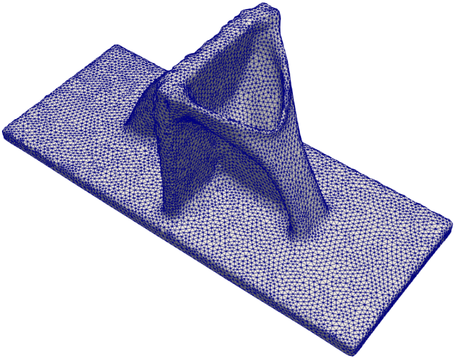


Figure: Optimized shape.

Fluid-structure interaction

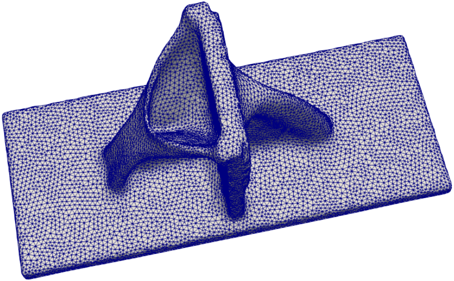


Figure: Optimized shape.

Fluid-structure interaction

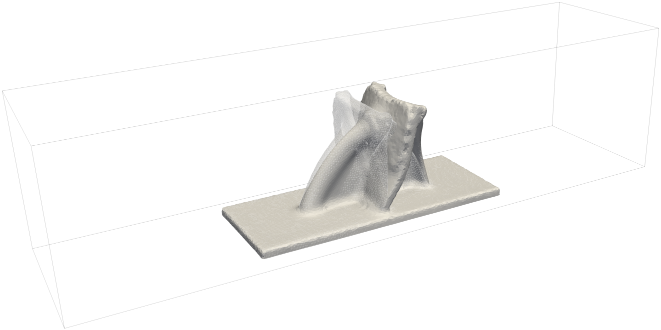
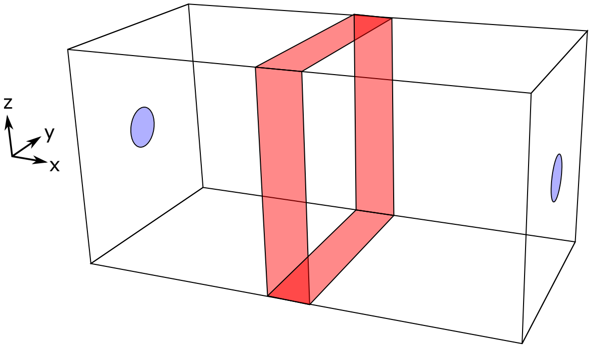


Figure: Elastic deformation.

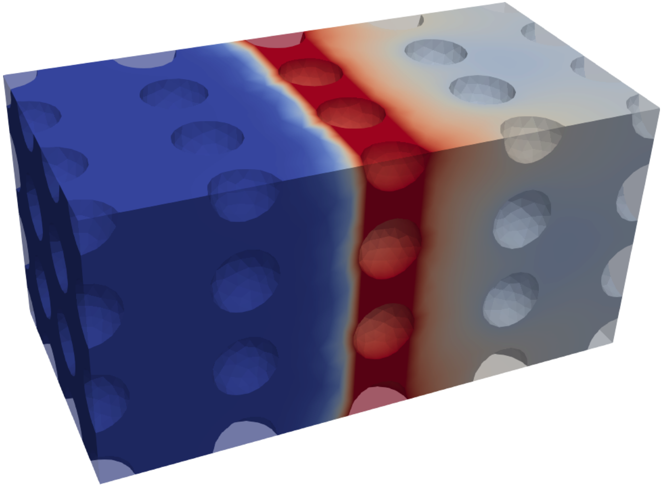
3D convective heat transfer



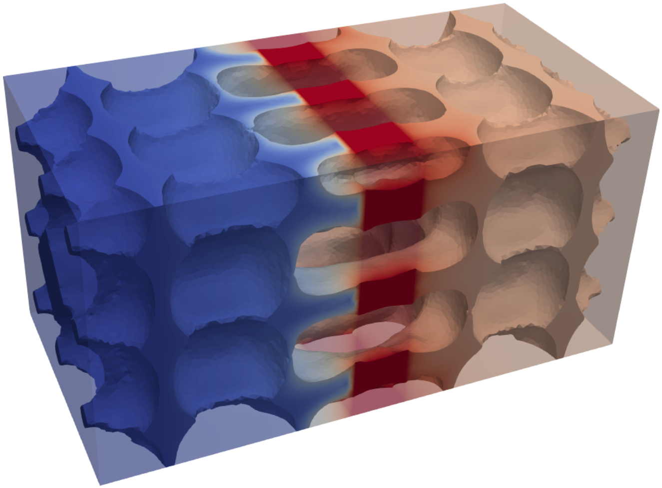
$$\min_{\Gamma} \quad J(\Gamma, \boldsymbol{v}(\Gamma), T(\Gamma)) := -\int_{\Omega_f} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x$$

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

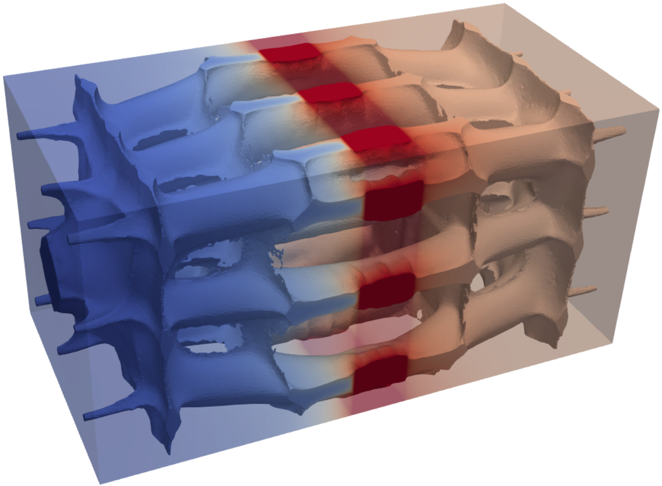
3D convective heat transfer



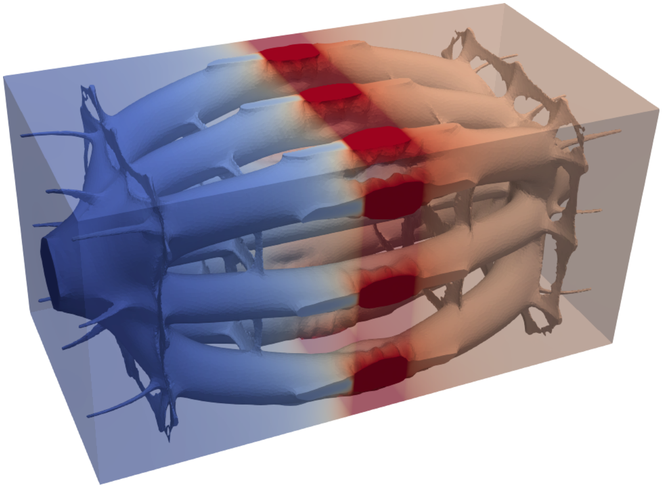
3D convective heat transfer



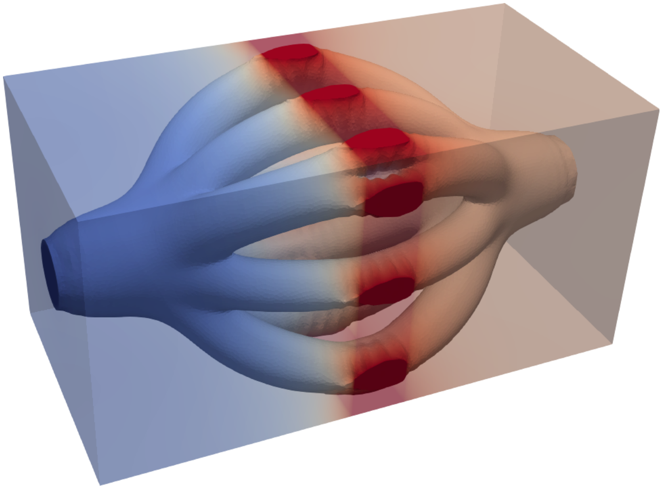
3D convective heat transfer



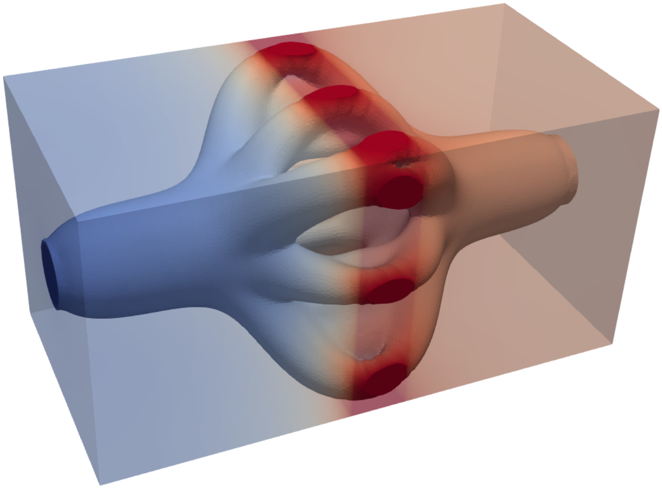
3D convective heat transfer



3D convective heat transfer



3D convective heat transfer



3D convective heat transfer

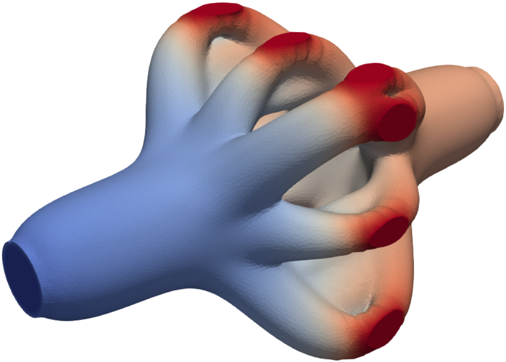


Figure: Optimized design.

3D convective heat transfer

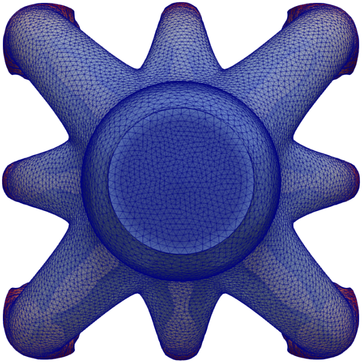


Figure: Optimized design.

3D convective heat transfer

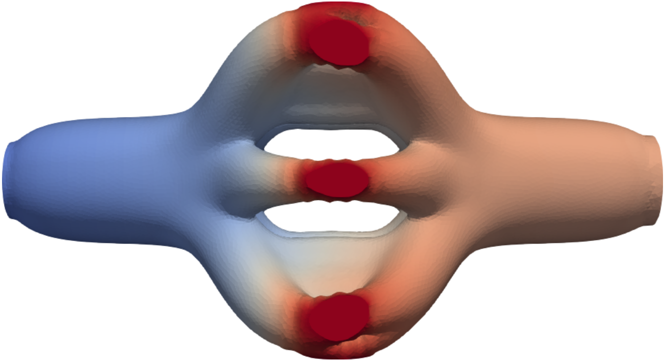


Figure: Optimized design.

3D convective heat transfer

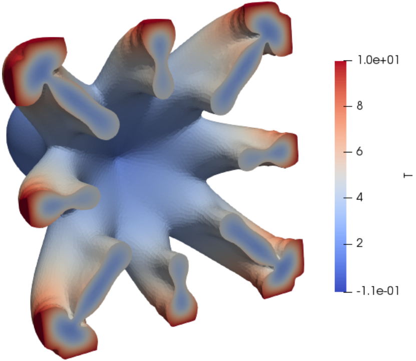


Figure: Optimized design.

Fluid-to-fluid heat exchangers

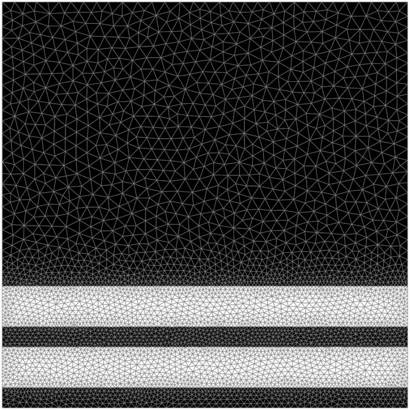
This allows to compute the shape derivatives of the heat transfer and of the pressure drop.

$$\max_{\Gamma = \overline{\Omega_f} \cap \overline{\Omega_s}} \qquad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$

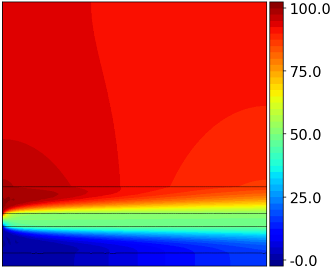
$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geqslant d_{\mathsf{min}}. \end{cases}$$

2D counter-current Heat exchanger

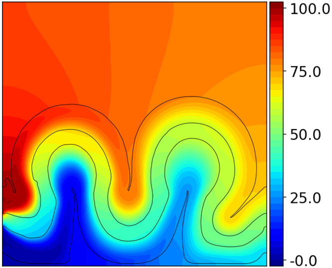
Iteration 0



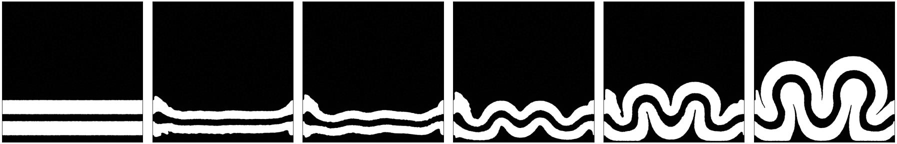
2D Heat Exchangers with non-mixing constraint



(a) Initial temperature



(b) Final temperature.



(c) Intermediate iterations 0, 8, 20, 50, 88 et 200.

2D Heat Exchangers with non-mixing constraint

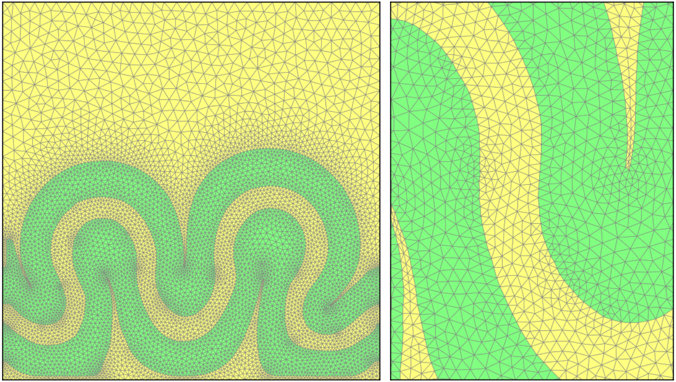
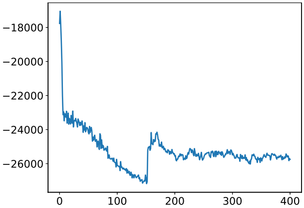
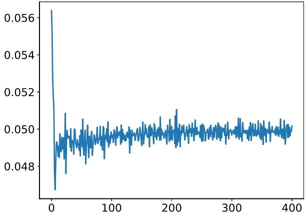


Figure: Zoom on the optimized mesh.

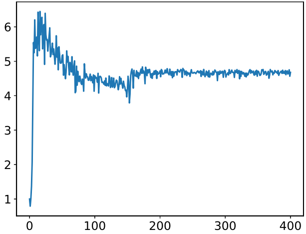
2D Heat Exchangers with non-mixing constraint



(a) Objective function $J(\Omega_f)$.



(b) Distance constraint $P_{\text{cold}\to\text{hot}}$ for the non-mixing constraint.



(c) Pressure drop constraint $\mathtt{DP}(\Omega_f)/\mathtt{DP}(\Omega_f^0)$.

3D fluid-to-fluid heat exchanger

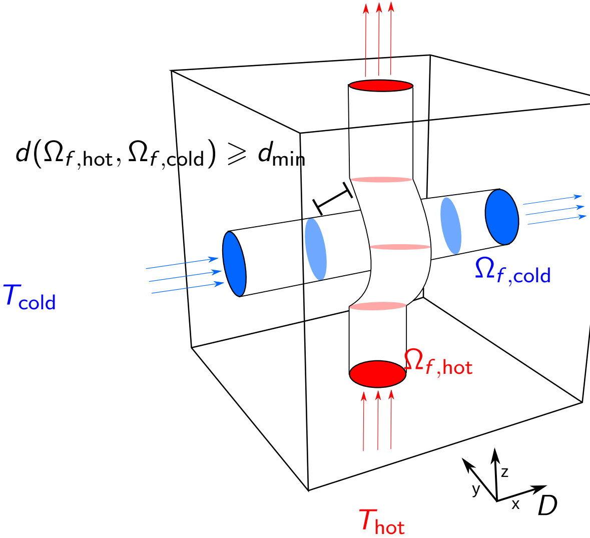


Figure: Schematic of the 3D setting.

3D fluid-to-fluid heat exchanger

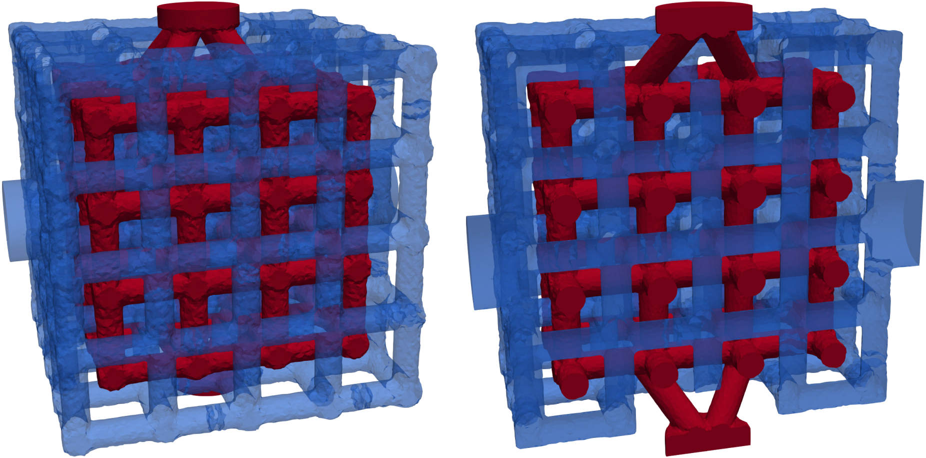
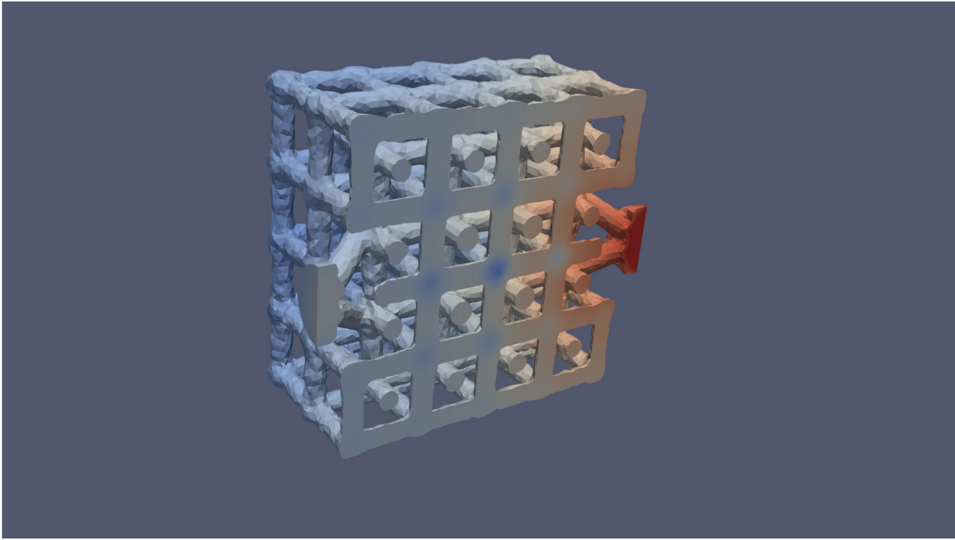
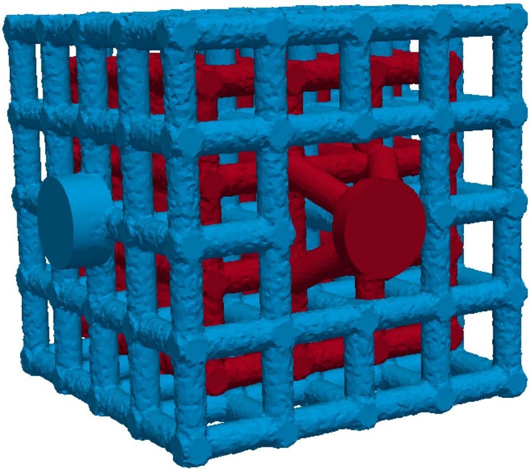


Figure: Initial distribution of fluid considered for the 3D heat exchanger test case.

3D fluid-to-fluid heat exchanger



3D fluid-to-fluid heat exchanger



3D fluid-to-fluid heat exchanger

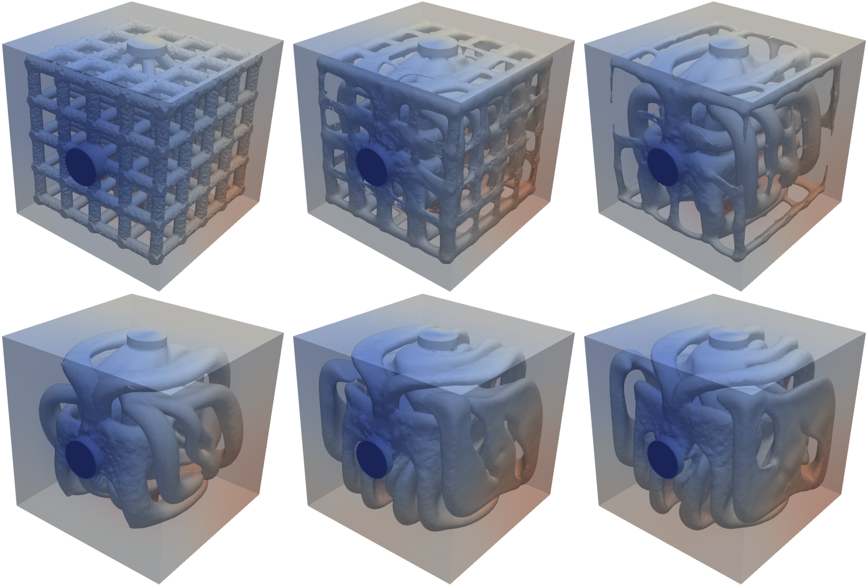
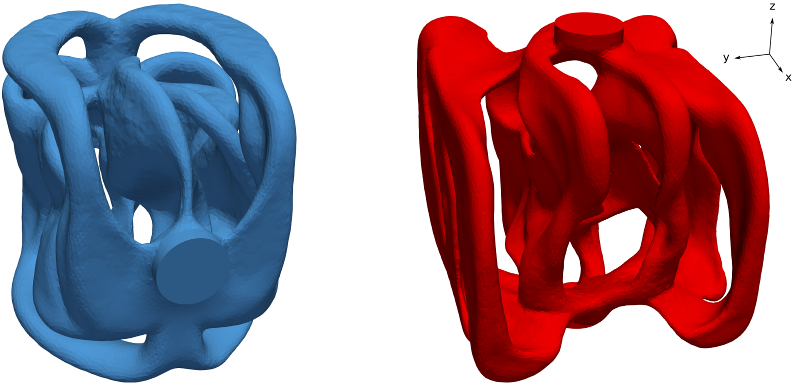


Figure: Intermediate iterations.

3D fluid-to-fluid heat exchanger



(a) Cold phase (b) Hot phase

Figure: Separate plots of the topologically optimized cold and hot fluid phases in the configuration $d_{\min} = 0.04$.

3D fluid-to-fluid heat exchanger

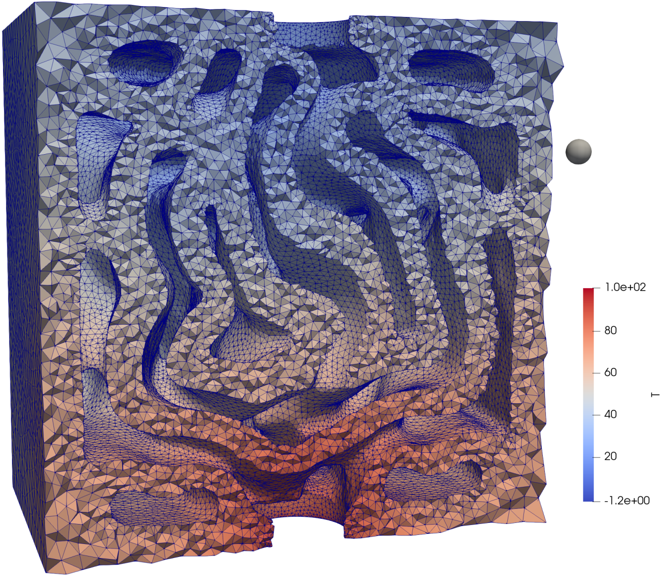
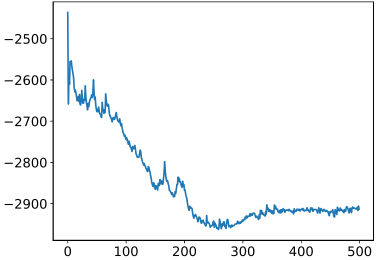
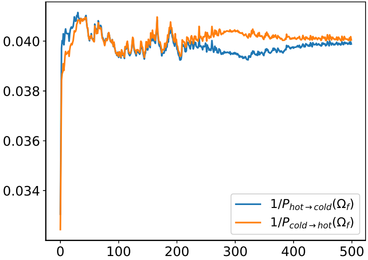


Figure: Cut of the resulting solid domain

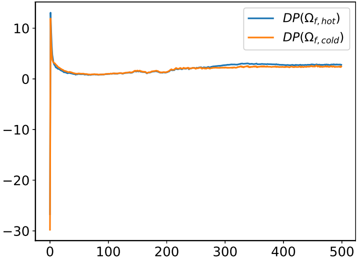
3D fluid-to-fluid heat exchanger



(a) Objective function (opposite of the heat exchanged).



(b) Averaged distance between the two phases.



(c) Pressure drop constraint.

Perspectives

- Change of physical model is possible in principle if

Perspectives

- Change of physical model is possible in principle if
    1. nonlinearities are differentiable

Perspectives

- Change of physical model is possible in principle if
  1. nonlinearities are differentiable
  2. adjoint equations (linearized transpose) can be solved numerically

Perspectives

- Change of physical model is possible in principle if
  1. nonlinearities are differentiable
  2. adjoint equations (linearized transpose) can be solved numerically

Perspectives

- Change of physical model is possible in principle if
  1. nonlinearities are differentiable
  2. adjoint equations (linearized transpose) can be solved numerically

  We expect turbulence can be treated in this manner.
- 3D remeshing is a bottle neck. Parallel remeshing will substantially reduce computational times.

Perspectives

- Change of physical model is possible in principle if
  1. nonlinearities are differentiable
  2. adjoint equations (linearized transpose) can be solved numerically

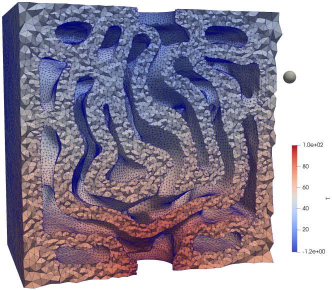
  We expect turbulence can be treated in this manner.
- 3D remeshing is a bottle neck. Parallel remeshing will substantially reduce computational times.
- Other topology optimization approaches, such as homogenization based, could lead to alternative methods for generating complex design.

Perspectives

- Change of physical model is possible in principle if
  1. nonlinearities are differentiable
  2. adjoint equations (linearized transpose) can be solved numerically

  We expect turbulence can be treated in this manner.
- 3D remeshing is a bottle neck. Parallel remeshing will substantially reduce computational times.
- Other topology optimization approaches, such as homogenization based, could lead to alternative methods for generating complex design.
- In contrast with density based methods, the body fitted approach allows to treat explicitly the non-mixing constraint, and is compatible in principle with non-intrusive solvers.

Many thanks for your attention!

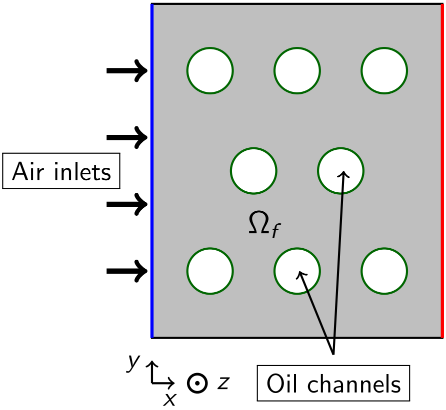


Appendix : an alternative, simple 2D heat exchanger model.

2D Air oil heat exchanger

Safran Aeroboosters case study:

$T = T_{oil}$ on Γ.
$T_{air} < T_{oil}$.



2D Air oil heat exchanger

- Optimization problem:

$$\begin{aligned} \min_{\Omega_f \subset D} \quad & J(\Omega_f) := -\int_{\Omega_f} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x \\ s.c. \quad & \mathtt{DP}(\Omega_f) := \int_{\partial\Omega_{f,in}} p \mathrm{d}s - \int_{\partial\Omega_{f,out}} p \mathrm{d}s \leq \mathtt{DP}_0. \end{aligned}$$

2D Air oil heat exchanger

- Optimization problem:

$$\min_{\Omega_f \subset D} \quad J(\Omega_f) := -\int_{\Omega_f} \rho c_p \boldsymbol{v} \cdot \nabla T \mathrm{d}x$$
$$s.c. \quad \mathtt{DP}(\Omega_f) := \int_{\partial\Omega_{f,in}} p \mathrm{d}s - \int_{\partial\Omega_{f,out}} p \mathrm{d}s \leq \mathtt{DP}_0.$$

- We consider an alternative formulation to impose a minimum thickness constraint on the oil channels.

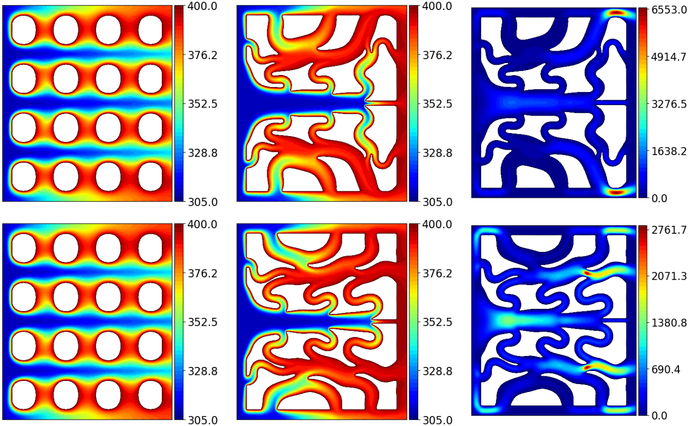
$$\min_{\Omega_f \subset D} \quad E(\Omega_f) := -\int_{D \setminus \Omega_f} d_{\Omega_f}^2 \max(-d_{\Omega_f} + d_{\min}/2, 0)^2 \mathrm{d}x$$
$$s.c. \quad \begin{cases} \mathtt{DP}(\Omega_f) \leq \mathtt{DP}_0 \\ J(\Omega_f) \leq \mathtt{J}_0. \end{cases}$$

2D Air oil heat exchanger

Results :

Test case 7
$||\boldsymbol{v}_0||_\infty = 10$
$DP_0 = 1300$
$J_{final} = 4086$
$DP_{final} = 1308$
Without minimum thickness.

Test case 8
$||\boldsymbol{v}_0||_\infty = 10$
$DP_0 = 1300$
$J_{final} = 4168$
$DP_{final} = 1188$
With minimum thickness.



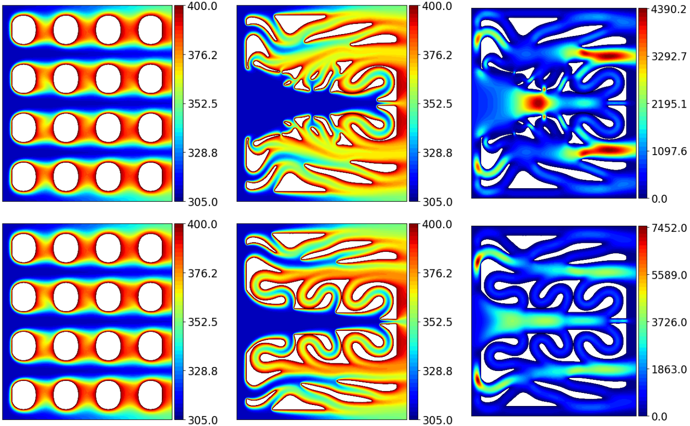
(a) T initial (b) Optimized T (c) Optimized $||\boldsymbol{v}||$

2D Air oil heat exchanger

Results :

Test case 9
$||\boldsymbol{v}_0||_\infty$=25
$\mathtt{DP}_0$=1030
J_{final} =7667
$\mathtt{DP}_{final}$ =968
Without minimum thickness.

Test case 10
$||\boldsymbol{v}_0||_\infty$=25
$\mathtt{DP}_0$=1030
J_{final} =7508
$\mathtt{DP}_{final}$ =1112
With minimum thickness.



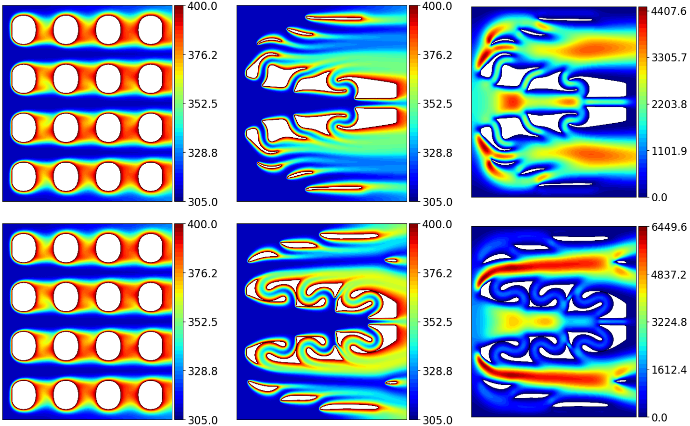
(a) T initial (b) Optimized T (c) Optimized $||\boldsymbol{v}||$

2D Air oil heat exchanger

Results :

Test case 11
$||\boldsymbol{v}_0||_\infty=40$
$\mathrm{DP}_0=475$
$J_{final}=5731$
$\mathrm{DP}_{final}=479$
Without minimum thickness.

Test case 12
$||\boldsymbol{v}_0||_\infty=40$
$\mathrm{DP}_0=475$
$J_{final}=6847$
$\mathrm{DP}_{final}=524$
With minimum thickness.



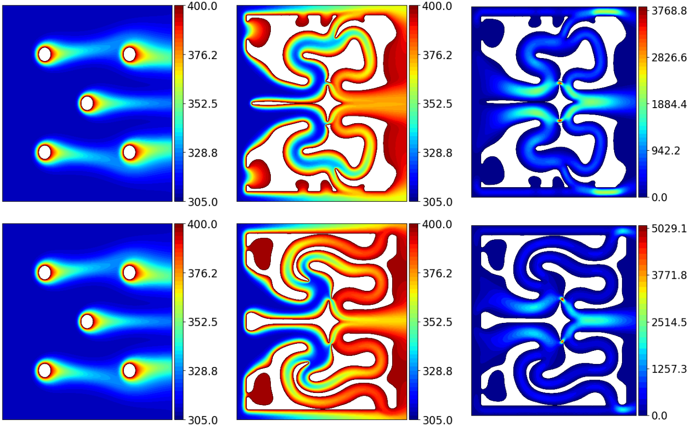
(a) T initial (b) Optimized T (c) Optimized $||\boldsymbol{v}||$

2D Air oil heat exchanger

Results :

Test case 16
$||\boldsymbol{v}_0||_\infty = 40$
$\mathrm{DP}_0 = 475$
$J_{final} = 5731$
$\mathrm{DP}_{final} = 479$
Without minimum thickness.

Test case 17
$||\boldsymbol{v}_0||_\infty = 40$
$\mathrm{DP}_0 = 475$
$J_{final} = 6847$
$\mathrm{DP}_{final} = 524$
With minimum thickness.



(a) T initial (b) Optimized T (c) Optimized $||\boldsymbol{v}||$