Shape and topology optimization applied to Compact Heat Exchangers

Florian Feppon

Shape and topology optimization applied to Compact Heat Exchangers

Florian Feppon *
ETH Zürich, Department of Mathematics

April 21st, 2021

Contents

*The material presented herein is based of my PhD research at École polytechnique, Palaiseau (France). This work was supported by the Association Nationale de la Recherche et de la Technologie (ANRT) [grant number CIFRE 2017/0024], by the project ANR-18-CE40-0013 SHAPO financed by the French Agence Nationale de la Recherche (ANR), and by a CIFRE funding from SAFRAN.

1 Introduction

The purpose of these notes is to offer a comprehensive introduction to topology optimization for automated generation of complex heat exchanger designs, based on the method of Hadamard whereby the design variable is the shape of the fluid-solid interfaces and is updated iteratively until convergence to a nearly optimal design. The material presented hereafter is intended to be an introductory exposure to our recent work (Feppon et al. (2021), illustrated on fig. 1) and PhD thesis (Feppon, 2019) where the reader is referred for further references and more detailed information about the theoretical ingredients of the methodologies presented hereafter.

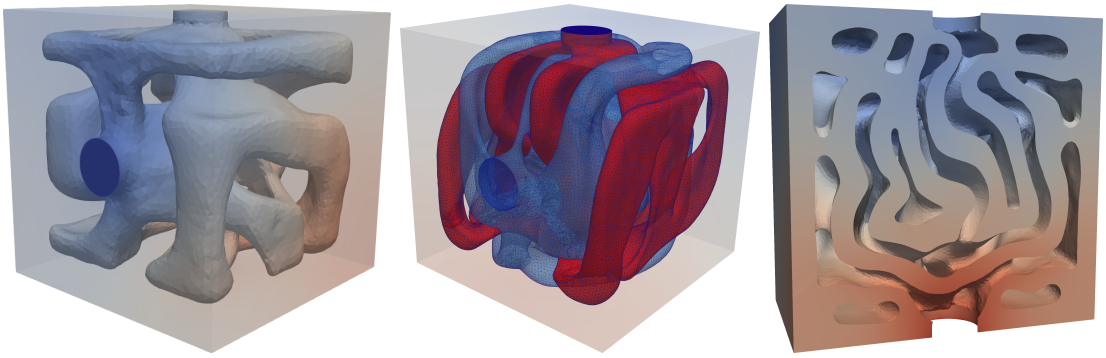
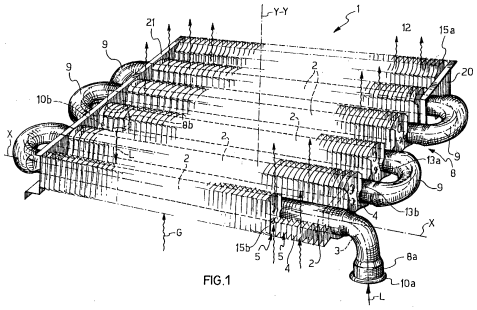
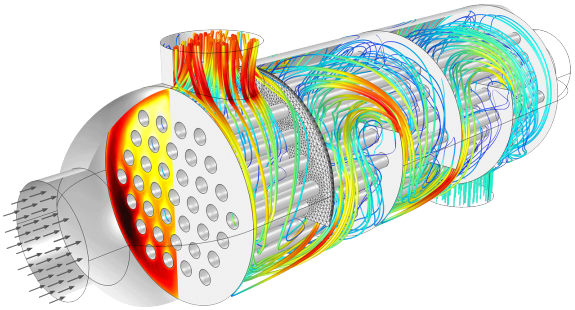


Figure 1: Topology optimized heat exchanger devices with the method of Hadamard and a body-fitted mesh evolution algorithm. Figures from Feppon et al. (2021).

Compact heat exchangers are devices whose task is to transfer heat from parts to parts of a system, usually between two distinct phases which can be either constituted of fluid, solid or gas, see e.g. Incropera et al. (2012) for an introduction. The phase entering the system with the lowest temperature is referred to as the "cold" phase, and the one entering with a higher temperature as the "hot" phase. Heat exchangers come into play in a broad range of industrial applications requiring cooling or heating gas, liquid or solid phases; they are involved e.g. in chemical engineering, combustion engines, radiators, refrigerants and heat sinks.



2.1: Industrial gas-liquid heat exchanger design featuring blade patterns (Figure from Giannoni (1999)).



2.2: Industrial air-water heat exchanger design featuring tube patterns (Figure from COMSOL multiphysics Multiphysics (1994)).

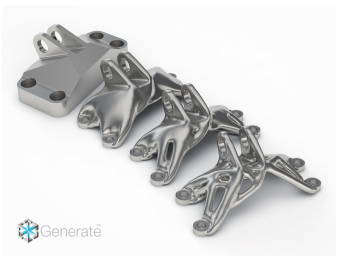
Figure 2: Two examples of industrial heat exchanger designs.

Heat exchangers must be designed in order to allow for optimal heat transfer: for this

reason, they are usually composed of an assembly of intricate tubes and fins devised to maximize the thermal exchange surface area (fig. 2). Simultaneously, they must meet a number of industrial specifications. The most common one is the need to moderate the output pressure loss due to the friction forces caused by solid walls on the fluid phases. In other words, the output debit should be the closest as possible to the one of the input phase. Mechanical constraints may also be needed to be considered, for instance to protect the device from the dilation due to elevated temperatures, or from damages caused by large pressure loads (see e.g. Capey (1965); Shah and Sekulic (2003); Laurent et al. (2016)). As it is common in most design problems, industrial specifications are competing between one another; for instance increasing the contact heat exchange surface area may also increase the amount of friction encountered by the fluid, thereby worsening the output pressure loss. Designing a heat exchanger, i.e. determining the shape of pipes, tubes and solid walls which account for all the specifications and allow optimal performance, is therefore a quite challenging task.

A common industrial practice is to rely on the assistance of computer-aided design (CAD) software, which enable engineers to find an optimal configuration of geometric variables parametrizing a proposed guess design (e.g. Bhutta et al. (2012); Awais and Bhuiyan (2018)). The main advantage of these software lies in their compatibility with all the stages of the industrial process, in particular with the commercial physical solvers used to evaluate the design properties or the with the manufacturing step (see for instance Harries et al. (2019); Duta et al. (2007); Dinh et al. (1996)). However, they yield by nature very little modifications of the initially proposed geometry, even though these may represent valuable gains of performance from an industrial point of view.

A rather recent trend in the industry is the increasing popularity of additive manufacturing which offers greater variety of manufacturable designs along with the possibility to use automated generative tools to devise them. The last twenty years have been marked by great progress in the development of the field of "topology optimization" and its variety of algorithms allowing to compute close to optimal designs free from any a priori on the geometry. Topology optimization is now a well established practice in various branches of the automotive, aeronautic or architecture industry (fig. 3), for the design of structures subjected to mechanical loads which are sought to be simultaneously lightweight and rigid. Many commercial software for numerical computations include today a topology optimization module for such purpose.



3.1: Siemens (2017)

3.2: APWorks (2016)

3.3: Airbus (2010)

Figure 3: Some applications of topology optimization of mechanical structures in the aeronautic and automotive industry.

Due to their ability to generate efficient and rather unconventional designs, topol-

ogy optimization algorithms offer promising perspectives for conceiving heat exchangers with substantial gain of performances. Such desired applications face however two main challenges.

First, the treatment of fluid systems involving potentially coupled *multiphysics constraints* such as convective heat transfer or fluid-structure interactions requires to handle more complex numerical and theoretical ingredients than for systems described uniquely by their mechanical properties. As a consequence, it is only very recently that research works have started to present complex 3D optimized designs for convective heat transfer (e.g. Yu et al. (2020) illustrated on fig. 4.1 or Pietropaoli et al. (2019); Huang et al. (2019); Feppon et al. (2020b)) and that a few commercial software include optimization capabilities for fluid systems, which remain to date rather limited.

Second, the design of heat exchangers featuring multiple fluid phases calls for the need to enforce a *non-mixing constraint* between the hot and cold input streams. The modelling and the implementation of this constraint into fluid topology optimization algorithms is delicate; it has been one of the most recent research trends in the field. The first published work on this topic is the seminal MsC thesis of Papazoglou (2015), followed since then by Saltzman et al. (2018); Haertel (2018); Saviers et al. (2019); Rebei (2019); Tawk et al. (2019); Kobayashi et al. (2020); Hoghoj et al. (2020).

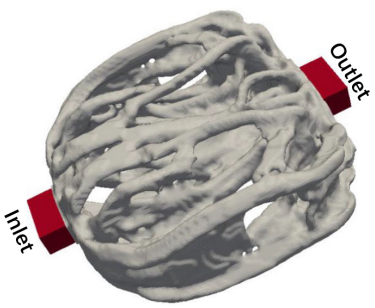
All the aforementioned works, some of them illustrated on fig. 4, rely on density based topology optimization methods: the design variable is a density field $\rho(x)$ which represents a proportion of fluid and solid at every location x of the design domain. The physical equations describing the system must be modified to encompass such relaxed porous states, and some penalization scheme be used in order to ensure the convergence of the design towards a state featuring only fluid ($\rho(x) = 1$) or solid ($\rho(x) = 0$). The exact location of the interface between the fluid domain and the solid phase is known only at convergence, which makes the modelling and the enforcement of the non-mixing constraint quite challenging, although possible.

In these lecture notes, we rather consider the method of Hadamard for the optimization of the interface between the solid and fluid phase. Its main advantage lies in that the location of the fluid-solid interface is known at all the stages of the optimization process. This enables to perform physical computations on body-fitted meshes in which the fluid-solid interface is discretized accurately, allowing in principle for non-intrusive simulations with external commercial software. Furthermore, this also allows to implement the non-mixing constraint between multiple fluid phases in a geometric manner: it reduces to impose a minimum solid wall thickness between each connected fluid components, which can be modelled as a minimum distance constraint.

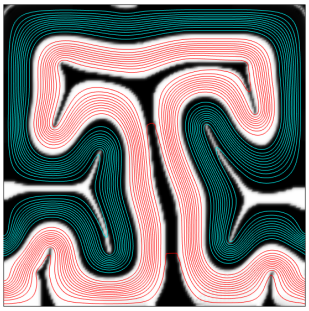
The implementation of a full topology optimization algorithm for multiple-phase heat-exchangers with the method of Hadamard requires the combination of multiple ingredients which are detailed in the next sections and summarized on fig. 5.

The first step is to settle a physical model and to recast the various industrial specifications into a mathematical optimization problem featuring objective and constraints, which is the object of section 2.

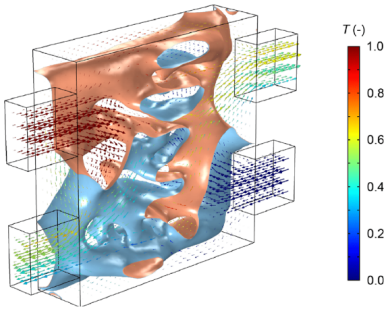
An introduction to the boundary variation method of Hadamard and to the notion of sensitivity with respect to variations of the shape is then provided in section 3. The key ingredient of the methodology is the knowledge of analytical formulas for the shape



4.1: Yu et al. (2020)

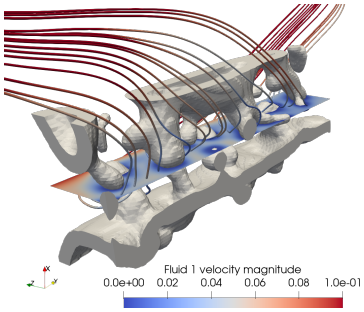


4.2: Papazoglou (2015)



4.3: Kobayashi et al. (2020)

4.4: Saviers et al. (2019)



4.5: Hoghoj et al. (2020)

Figure 4: Fluid pipes optimized for convective heat transfer with density-based methods.

derivatives of quantities involved in convective-heat transfer problems, which are stated without proof. Finally, we discuss the additional ingredients needed for the treatment of the geometric non-mixing constraint in the context of multiple phases heat exchangers.

The next section 4 is concerned with numerical gradient-based optimization algorithms whose main task is to convert the shape sensitivities into a design update which improves performance while gradually enforcing the constraints. We provide a quick introduction to the null-space gradient flow algorithm, and to how to interface it with a level set based mesh evolution method for the numerical shape updates. The latter enables to handle complex topological changes in the course of the optimization process while still maintaining an explicit discretization of shapes suitable for numerical fluid computations. The overall methodology is then summarized and numerical illustrations are provided on heat exchanger test cases.

We provide some concluding remarks and perspectives in section 5.

2 Formulation of the optimal heat exchanger design problem

2.1 Physical modelling

The first step of the design methodology is to agree on a physical model describing the system in a way which is as realistic as possible while still allowing for numerical computations and optimization. It is already by nature a rather complex task, due to the variety of possible models depending on the desired working conditions for the device, on

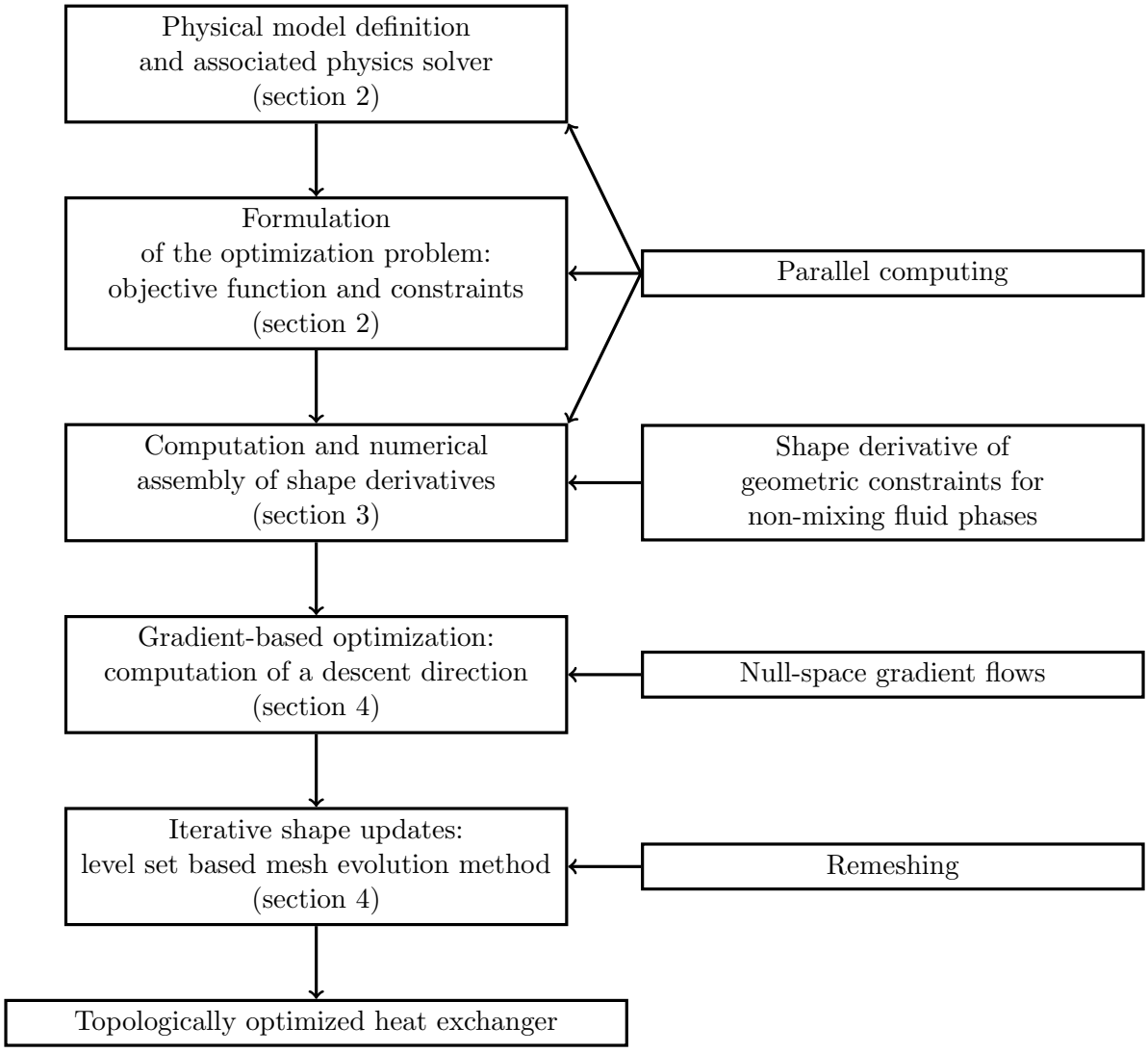


Figure 5: Topology optimization workflow for automated design of multiple-phase heat exchangers.

the chosen materials, on the need for taking into account non-linear effects due to high temperature or turbulence due to high Reynolds numbers.

For our applications, we consider a rather generic situation where the system occupies a box $D = \overline{\Omega_f} \cup \overline{\Omega_s} \subset \mathbb{R}^d$ with $d = 2, 3$ constituted of fluid components Ω_f and a solid component Ω_s. The fluid-solid interface is denoted by $\Gamma := \overline{\Omega_f} \cap \overline{\Omega_s}$. The fluid part $\Omega_f = \Omega_{f,\text{hot}} \cup \Omega_{f,\text{cold}}$ is itself constituted of hot and cold phases $\Omega_{f,\text{hot}}$ and $\Omega_{f,\text{cold}}$ which are characterized by the same density ρ, dynamic viscosity ν, heat conductivity k_f and heat capacity c_p. The solid phase is assumed to be made of a single material of conductivity k_s. The setting is illustrated on fig. 6.

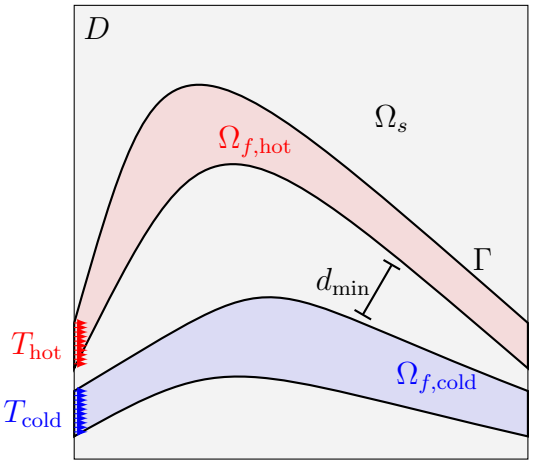


Figure 6: Setting of the heat exchanger topology optimization problem (not all boundaries are represented).

The fluid enters the system at the boundary $\partial\Omega_{f,\text{in}}$ with a velocity profile $\boldsymbol{v}_0$ and exits it with zero normal stress on the output boundary $\partial\Omega_{f,\text{out}}$. The fluid sticks to the wall and therefore its velocity vanishes on the solid-interface boundary Γ. The motion of the fluid is described by its velocity and pressure variables $(\boldsymbol{v}, p)$. For our applications, we assume that these are solution to the steady-state incompressible Navier-Stokes equations:

$$\begin{cases} -\text{div}(\sigma_f(\boldsymbol{v}, p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = 0 & \text{in } \Omega_f \\ \text{div}(\boldsymbol{v}) = 0 & \text{in } \Omega_f \\ \boldsymbol{v} = \boldsymbol{v}_0 & \text{on } \partial\Omega_{f,\text{in}} \\ \sigma_f(\boldsymbol{v}, p)\boldsymbol{n} = 0 & \text{on } \partial\Omega_{f,\text{out}} \\ \boldsymbol{v} = 0 & \text{on } \Gamma, \end{cases} \tag{1}$$

where the fluid stress tensor $\sigma_f(\boldsymbol{v}, p)$ is given by Newton's law:

$$\sigma_f(\boldsymbol{v}, p) = 2\nu e(\boldsymbol{v}) - p\text{I}, \text{ with } e(\boldsymbol{v}) := \frac{1}{2}(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T), \tag{2}$$

with I is the identity 2×2 matrix and $\nabla \boldsymbol{v} = (\partial_j v_i)_{1 \le i,j \le d}$ the Jacobian matrix.

The velocity field $\boldsymbol{v}$ determines then the temperature distribution in both fluid and solid components through convection (in the fluid phase) and diffusion (in both solid and fluid phases). The cold and hot fluid phases enter the domain with given temperatures

$T_{\text{cold}} < T_{\text{hot}}$ on their respective inlet boundaries. It is then assumed that the temperature field T is continuous at the solid-fluid interface Γ as well as the heat flux. All other boundaries are adiabatic (no heat loss). We further assume that the temperature field T satisfies the steady-state convection-diffusion equation. Denoting by T_s and T_f the restrictions of T to respectively the solid and the fluid subdomains Ω_s and Ω_f, these read

$$\begin{cases} -\text{div}(k_f \nabla T_f) + \rho c_p \boldsymbol{v} \cdot \nabla T_f = 0 & \text{in } \Omega_f \\ -\text{div}(k_s \nabla T_s) = 0 & \text{in } \Omega_s \\ T = T_{\text{hot}} & \text{on } \partial\Omega_{f,\text{in}} \cap \partial\Omega_{f,\text{hot}} \\ T = T_{\text{cold}} & \text{on } \partial\Omega_{f,\text{in}} \cap \partial\Omega_{f,\text{cold}} \\ -k_f \dfrac{\partial T_f}{\partial \boldsymbol{n}} = 0 & \text{on } \partial\Omega_{f,\text{out}} \\ -k_s \dfrac{\partial T_s}{\partial \boldsymbol{n}} = 0 & \text{on } \partial\Omega_s \backslash \Gamma \\ T_f = T_s & \text{on } \Gamma \\ -k_f \dfrac{\partial T_f}{\partial \boldsymbol{n}} = -k_s \dfrac{\partial T_s}{\partial \boldsymbol{n}} & \text{on } \Gamma. \end{cases} \tag{3}$$

Note that both systems of equations eqs. (1) and (3) are solved successively and independently. Change of boundary conditions or of the physical equations may be considered without altering fundamentally the topology optimization methodology exposed hereafter. The physical variables $(\boldsymbol{v}, p)$ and T depend on the shape of the fluid domain Ω_f and we shall sometimes emphasize this dependence explicitly by writing e.g. $\boldsymbol{v} \equiv \boldsymbol{v}(\Omega_f)$.

A numerical topology optimization algorithm requires eqs. (2) and (3) to be solved many times during the total optimization procedure, typically from fifty to a few hundreds of times. Although we shall delve into the details regarding the many numerical methods available for this task, it is important to emphasize that efficient and accurate solvers are needed for such purpose, especially to tackle 3D systems. In particular, it is especially beneficial to resort to domain decomposition methods and efficient physics based preconditioners, see e.g. Feppon et al. (2020b) for further discussions based on this work and the references therein.

2.2 Heat exchanged and maximal pressure drop

Once the physical model has been mathematically settled, industrial specifications needs to be translated in terms of mathematical constraints on quantities which can be computed from the state variables $(\boldsymbol{v}, p)$ and T introduced above. In these notes, we consider two physical quantities commonly considered by engineers when it comes to characterize the performance of a heat exchanger. These are respectively the exchanged heat or *output power* and the *output pressure loss*.

The *exchanged heat* between both components $\Omega_{f,\text{hot}}$ and $\Omega_{f,\text{cold}}$ can be mathematically modelled by the quantity $\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$ defined by

$$\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f)) = \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T\, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T\, \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y, \tag{4}$$

which measures the difference between the heat fluxes exiting the hot phase and the one exiting the cold phase. It is then expected that the quantity $\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$ is maximized for configurations characterized by a small amount of heat exiting the hot fluid output and a large one from the cold fluid output, meaning that the hot fluid has been cooled down while the cold fluid has been heated up by receiving the heat of the hot phase.

The *output pressure drop* associated to the cold channel (respectively the hot channel) is modelled by the quantity $\mathtt{DP}(\Omega_{f,\text{cold}}, p(\Omega_f))$ (respectively $\mathtt{DP}(\Omega_{f,\text{hot}}, p(\Omega_f))$) defined by:

$$\mathtt{DP}(\Omega_{f,\text{cold}}, p(\Omega_f)) := \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{in}}} p \,\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}\cap\partial\Omega_{f,\text{out}}} p \,\mathrm{d}y \tag{5}$$

where $p \equiv p(\Omega_f)$ is the pressure inside the fluid given by the Navier-Stokes equation eq. (1). The output pressure drop is related to the loss of flow rate between the output and the input, or to the loss of energy due to the friction incurred by the contact with the walls. Alternative definitions of eq. (5) are possible, for instance in terms of the dynamic pressure $p + \rho v^2/2$ rather than the static pressure p, or by considering the strain energy $\int_{\Omega_{f,\text{cold}}} \nabla\boldsymbol{v} : \nabla\boldsymbol{v}\mathrm{d}x$ instead of the pressure drop.

A heat exchanger should have low output pressure drops in order to avoid slowing down substantially each of the input fluid phases, in view of being integrated properly as part of a larger thermal-hydraulic system.

2.3 Non-mixing constraint

An implicit requirement of multiple phase heat exchangers is that the hot stream and the cold stream should not mix in order to allow for later use of both phases; if mixing was allowed then designing heat exchangers would be trivial! Since designs in which both phases merge are certainly optimal for heat transfer, it is necessary to formulate the non-mixing constraint mathematically for it to be taken into account by automated design algorithms.

A convenient way to formulate the constraint is to state that a minimum solid wall thickness should be imposed between the hot and the cold phases (illustrated on fig. 6). Mathematically, this is equivalent to require that the distance between the boundaries of the hot and cold phase be greater than a desired value $d_{\min}$:

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

2.4 Formulation of the optimal design problem

Considering the modelling equations eqs. (1) and (3), the optimal design problem for a two-fluid heat exchanger accounting for the desire to maximize the heat exchange $\mathtt{W}$, to maintain a low output pressure drop while keeping both input and output phases well-

separated can be formulated as the following optimization program:

$$\max_{\Gamma=\overline{\Omega_f}\cap\overline{\Omega_s}} \quad \mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$$
$$s.t. \begin{cases} \mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ \mathtt{DP}(\Omega_{f,\mathrm{hot}}, p(\Omega_f)) \leq \mathtt{DP}_0 \\ d(\Omega_{f,\mathrm{cold}}, \Omega_{f,\mathrm{hot}}) \geq d_{\min}. \end{cases} \tag{6}$$

where $\mathtt{DP}_0$ is a given desired upper bound for the allowed maximum pressure drop.

Depending on the applications, additional constraints may be considered such as a maximum value Vol_0 allowed for the mass of each fluid phases passing through the heat exchanger:

$$\mathtt{Vol}(\Omega_{f,\mathrm{cold}}) \leq \mathtt{Vol}_0, \quad \mathtt{Vol}(\Omega_{f,\mathrm{hot}}) \leq \mathtt{Vol}_0. \tag{7}$$

The constraints of eq. (7) are not mandatory, but they are useful for instance if one desires that only a limited proportion of fluid of the total system lies in the heat exchanger at a given instant. Equation (7) considers one constraint for each phase in order to ensure that these remain balanced in terms of volume.

The next sections detail a topology optimization method for solving the optimization program of eq. (6).

3 Shape and Topology optimization with the method of Hadamard

In order to solve the nonlinear, non-parametric constrained optimization program of eq. (6), we consider a topology optimization procedure based on the boundary variation method of Hadamard. The main idea is to solve eq. (6) by applying iteratively small deformations $\mathrm{I} + \boldsymbol{\theta}$ of the fluid-solid interface Γ (illustrated on fig. 7) which gradually improve the design. The small vector field $\boldsymbol{\theta}$ is found as a descent direction by a suitable gradient based optimization algorithm. This is possible by computing the sensitivities of the objective and constraint functions $\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$ and $\mathtt{DP}(\Omega_{f,\mathrm{cold}}, p(\Omega_f))$ with respect to the variations of the shape of Ω_f, or equivalently of the fluid-solid interface Γ.

By essence, the method of Hadamard, which rests upon gradual shape deformations, yields a priori *shape* optimization algorithms. However, when the numerical update of shapes allows topological changes to occur, e.g. when resorting to implicit level set descriptions (fig. 8), we refer to it as a *topology optimization method*. The method is now well established and there is a large literature concerned with both theoretical and numerical aspects, see e.g. Sokolowski and Zolésio (1992); Henrot et al. (2018) and Allaire et al. (2021) for a recent introductory survey to the field. In what follows, we provide a brief summary to the notion of shape derivative which is the most important theoretical ingredient for the purpose of solving eq. (7). This part focuses mostly on the theoretical notion of shape derivative, leaving the matter of numerical shape updates as part of a global optimization procedure to the next section 4.

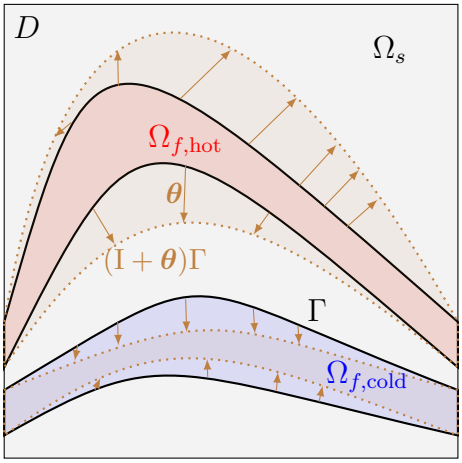
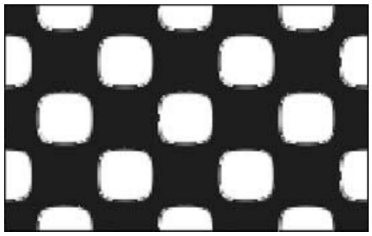
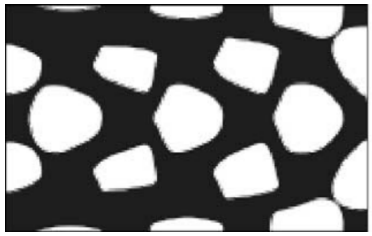


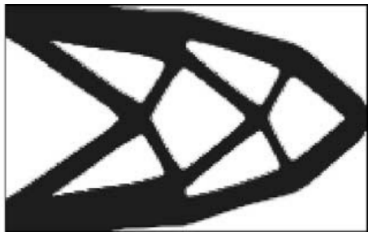
Figure 7: The boundary variation method of Hadamard, which considers small shape deformations $\boldsymbol{\theta}$ of the fluid-solid interface Γ. The new boundary $(\mathrm{I}+\boldsymbol{\theta})\Gamma$ is visible in dashed line.



8.1: Initial design



8.2: Intermediate design



8.3: Final design

Figure 8: Topology optimization of a 2D mechanical structure based on Hadamard's shape sensitivities. The level set method for the numerical evolution of shapes allows topological changes to occur such as the merging of holes (figures from Allaire et al. (2002)).

3.1 Hadamard shape derivatives

Let $F(\Omega_f)$ be a quantity depending on the shape of the fluid domain Ω_f (equivalently, of the solid domain Ω_s or on the interface Γ). For a given small vector field $\boldsymbol{\theta}$, $F((\mathrm{I}+\boldsymbol{\theta})\Omega_f)$ is the value of this quantity on the deformed configuration $(\mathrm{I}+\boldsymbol{\theta})\Omega_f$ (we assume that $\boldsymbol{\theta}=0$ on the boundaries which are not the fluid-solid interface Γ). The main idea lying behind the definition of shape derivative rests on the fact that considering small deformation vector fields $\boldsymbol{\theta}$ enables one to consider first order asymptotic expansions of the quantity $F((\mathrm{I}+\boldsymbol{\theta})\Omega_f)$ with respect to $\boldsymbol{\theta}$.

In this spirit, the functional $\Omega_f \mapsto F(\Omega_f)$ is said to be shape differentiable if there exists a linear mapping $\mathrm{D}F(\Omega_f)$ such that the following first-order expansion holds at $\boldsymbol{\theta}=0$:

$$F((\mathrm{I}+\boldsymbol{\theta})(\Omega_f)) = F(\Omega_f) + \mathrm{D}F(\Omega_f)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}) \text{ where } \frac{o(\boldsymbol{\theta})}{||\boldsymbol{\theta}||_V} \xrightarrow{\boldsymbol{\theta}\to 0} 0, \tag{8}$$

where the smallness of $o(\boldsymbol{\theta})$ is measured with the norm $||\cdot||_V$ of some (Banach or Hilbert) space V. The linear map $\mathrm{D}F(\Omega_f)$ is called the shape derivative of the functional F.

3.2 Shape derivatives formulas for heat exchanger design

The practical computation of $\mathrm{D}F(\Omega_f)$ when $F(\Omega_f) \equiv F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))$ depends on the solution to some partial equations (such as in eq. (6)) is achieved thanks to shape calculus (Delfour and Zolesio (2011); Henrot et al. (2018)).

The most general expression of $\mathrm{D}F(\Omega_f)$ involves volume integrals of quantities depending on the state variables $(\boldsymbol{v}, p)$ and T, their respective *adjoint states* $(\boldsymbol{w}, q)$ and S, and the values of the *gradient* of the vector field $\boldsymbol{\theta}$ on the design domain D. In Feppon et al. (2019a), the following expression has been derived:

$$\begin{aligned}
&\mathrm{D}F(\Omega_f, \boldsymbol{v}(\Omega_f), p(\Omega_f), T(\Omega_f))(\boldsymbol{\theta}) \\
&\quad = \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) - \int_{\Omega_f} (\sigma_f(\boldsymbol{v},p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w}\cdot\nabla \boldsymbol{v}\,\boldsymbol{v})\mathrm{div}(\boldsymbol{\theta})\mathrm{d}x \\
&\quad + \int_{\Omega_f} [\sigma_f(\boldsymbol{v},p) : (\nabla \boldsymbol{w}\nabla\boldsymbol{\theta}) + \sigma_f(\boldsymbol{w},q) : (\nabla \boldsymbol{v}\nabla\boldsymbol{\theta}) + \rho \boldsymbol{w}\cdot(\nabla \boldsymbol{v}\nabla\boldsymbol{\theta})\,\boldsymbol{v}]\mathrm{d}x \\
&\quad - \int_{\Omega_s} \mathrm{div}(\boldsymbol{\theta})(k_s \nabla T\cdot\nabla S)\mathrm{d}x - \int_{\Omega_f} \mathrm{div}(\boldsymbol{\theta})(k_f \nabla T\cdot\nabla S + \rho c_p(\boldsymbol{v}\cdot\nabla T)S)\mathrm{d}x \\
&\quad + \int_{\Omega_s} k_s(\nabla\boldsymbol{\theta} + \nabla\boldsymbol{\theta}^T)\nabla T\cdot\nabla S\mathrm{d}x \\
&\quad + \int_{\Omega_f} \left[k_f(\nabla\boldsymbol{\theta} + \nabla\boldsymbol{\theta}^T)\nabla T\cdot\nabla S + \rho c_p \boldsymbol{v}\cdot(\nabla\boldsymbol{\theta}^T\nabla T)S\right]\mathrm{d}x,
\end{aligned} \tag{9}$$

where $\partial F/\partial\boldsymbol{\theta}$ is a certain partial derivative of F with respect to the variations of Ω_f. Equation (9) is referred to as the *volume* expression of the shape derivative. As complicated it might look, it can be implemented numerically once for all in a numerical optimization procedure, since this general expression is valid for *arbitrary* objective or constraint functionals F.

The adjoint states $(\boldsymbol{w}, p)$ and S are found by solving a linearized and transposed version of the state equations eqs. (1) and (3), where the right-hand side are the partial

derivatives of F with respect to the state variables. The adjoint system of equations is given in its most general version in the following variational form: find $(\boldsymbol{w}, q) \in V_{\boldsymbol{v},p}$ and $S \in V_T$ such that

$$\int_{\Omega_s} k_s \nabla S \cdot \nabla S' \mathrm{d}x + \int_{\Omega_f} (k_f \nabla S \cdot \nabla S' + \rho c_p S \boldsymbol{v} \cdot \nabla S') \mathrm{d}x = \frac{\partial F}{\partial T}(S) \quad \forall S' \in V_T \,. \tag{10}$$

$$\begin{aligned} \forall (\boldsymbol{w}', q') \in V_{\boldsymbol{v},p} \\ \int_{\Omega_f} \Big(\sigma_f(\boldsymbol{w}, q) : \nabla \boldsymbol{w}' + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{w}' \cdot \boldsymbol{v} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v} \cdot \boldsymbol{w}' - q' \mathrm{div}(\boldsymbol{w}) \Big) \mathrm{d}x = \\ \int_{\Omega_f} -\rho c_p S \nabla T \cdot \boldsymbol{w}' \mathrm{d}x + \frac{\partial F}{\partial (\boldsymbol{v}, p)}(\boldsymbol{w}', q'), \end{aligned} \tag{11}$$

where $V_{\boldsymbol{v},p}$ and V_T are variational spaces naturally associated to the variables $(\boldsymbol{v}, p)$ and T, and which encompass all Dirichlet boundary condition (i.e. $\boldsymbol{w} = 0$ on $\Gamma \cup \partial\Omega_{f,\mathrm{in}}$ and $S = 0$ on $\partial\Omega_{f,\mathrm{in}}$). Note that the adjoint fluid equation eq. (11) is solved after the adjoint thermal equation eq. (10), which is the reversed order of the state equations eqs. (1) and (3). Similarly, eqs. (10) and (11) can be implemented once for all in the optimization routine and be solved with the finite element method using domain decomposition and parallel computing.

Under suitable regularity assumptions, Hadamard's structure theorem predicts that the shape derivative $\mathrm{D}F(\Omega_f)(\boldsymbol{\theta})$ should depend only on the normal component $\boldsymbol{\theta} \cdot \boldsymbol{n}$ on the deformed interface Γ. Under this circumstance, it is possible after some integration by parts to rewrite $\mathrm{D}F(\Omega_f)$ in the form of a boundary integral depending only on $\boldsymbol{\theta} \cdot \boldsymbol{n}$:

$$\mathrm{D}F(\Omega_f)(\boldsymbol{\theta}) = \int_\Gamma v_F(\Omega_f) \boldsymbol{\theta} \cdot \boldsymbol{n} \mathrm{d}y, \tag{12}$$

for some function $v_F(\Omega_f)$ defined on Γ. Equation (12) is referred to as the "surface expression" of the shape derivative. In our heat exchanger context, the function $v_F(\Omega_f)$ is given explicitly by the formula

$$\begin{aligned} v_F(\Omega_f) =& \boldsymbol{g}_F(\Omega_f) \cdot \boldsymbol{n} - \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \boldsymbol{n} \cdot (\sigma_f(\boldsymbol{w}, q) \nabla \boldsymbol{v}) \boldsymbol{n} + \boldsymbol{n} \cdot (\sigma_f(\boldsymbol{v}, p) \nabla \boldsymbol{w}) \boldsymbol{n} \\ &+ k_s \nabla T_s \cdot \nabla S_s - k_f \nabla T_f \cdot \nabla S_f - 2 k_s \frac{\partial T_s}{\partial \boldsymbol{n}} \frac{\partial S_s}{\partial \boldsymbol{n}} + 2 k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \frac{\partial S_f}{\partial \boldsymbol{n}} \end{aligned}$$

where $\boldsymbol{g}_F(\Omega_f)$ is the unique function such that

$$\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = \int_D \boldsymbol{f}_F(\Omega_f) \cdot \boldsymbol{\theta} \mathrm{d}x + \int_\Gamma \boldsymbol{g}_F(\Omega_f) \cdot \boldsymbol{\theta} \mathrm{d}y$$

for some vector field $\boldsymbol{f}_F(\Omega_f)$, obtained e.g. thanks to an integration by part. In part of the literature, the function $v_F(\Omega_f)$ is sometimes also called the "shape derivative" of F, while the distinction with the linear form $\mathrm{D}F(\Omega_f)$ has important numerical consequences. Indeed, several works have noticed at least empirically (see e.g. Hiptmair et al. (2015)) that the volume expression eq. (9) leads to more accurate numerical evaluations of the

shape derivative, leading to better designs. We also observed this fact in our own numerical applications, and hence we favour the use of the volume expression of eq. (9) of the shape derivative rather than the surface expression of eq. (12).

Coming back to the optimization program of eq. (6), the previous formulas enable one to compute the shape derivatives of the heat exchanged $\mathtt{W}(\Omega_f, \boldsymbol{v}(\Omega_f), T(\Omega_f))$ and of the pressure drop $\mathtt{DP}(\Omega_{f,\text{cold}}, p(\Omega_f))$. The only missing ingredient are the partial derivatives involved in eq. (9) and the adjoint equations eqs. (10) and (11), which are easily computed:

$$\frac{\partial \mathtt{W}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) = 0, \quad \frac{\partial \mathtt{W}}{\partial (\boldsymbol{v}, p)}(\boldsymbol{w}', q') = \int_{\partial\Omega_{f,\text{hot}}} \rho c_p T \boldsymbol{w}' \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p T \boldsymbol{w}' \cdot \boldsymbol{n} \mathrm{d}y$$

$$\frac{\partial \mathtt{W}}{\partial T}(S') = \int_{\partial\Omega_{f,\text{hot}}} \rho c_p S' \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}} \rho c_p S' \boldsymbol{v} \cdot \boldsymbol{n} \mathrm{d}y.$$

$$\frac{\partial \mathtt{DP}}{\partial \boldsymbol{\theta}}(\Omega_{f,\text{cold}}, p(\Omega_f))(\boldsymbol{\theta}) = 0, \quad \frac{\partial \mathtt{DP}}{\partial T}(\Omega_{f,\text{cold}}, p(\Omega_f))(S') = 0,$$

$$\frac{\partial \mathtt{DP}}{\partial (\boldsymbol{v}, p)}(\Omega_{f,\text{cold}}, p(\Omega_f))(\boldsymbol{w}', q') = \int_{\partial\Omega_{f,\text{cold}} \cap \partial\Omega_{f,\text{in}}} q' \mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}} \cap \partial\Omega_{f,\text{out}}} q' \mathrm{d}y.$$

3.3 Shape sensitivity of geometric constraints

The computation of the shape derivative of the non-mixing constraint

$$d(\Omega_{f,\text{cold}}, \Omega_{f,\text{hot}}) \geq d_{\min} \tag{13}$$

of eq. (6) is not standard and requires a particular treatment. It is convenient to first reformulate it in terms of the signed distance function $d_{\Omega_{f,\text{hot}}}$ associated to the hot fluid component. The latter is equal to minus plus the distance to the boundary $\partial\Omega_{f,\text{hot}}$ on respectively the interior and the complement of the domain $\Omega_{f,\text{hot}}$:

$$\forall x \in D,\ d_{\Omega_{f,\text{hot}}}(x) = \begin{cases} -\min\limits_{y \in \partial\Omega_{f,\text{hot}}} |x - y| \text{ if } x \in \Omega_{f,\text{hot}} \\ 0 \text{ if } x \in \partial\Omega_{f,\text{hot}} \\ \min\limits_{y \in \partial\Omega_{f,\text{hot}}} |x - y| \text{ if } x \in D \setminus \Omega_{f,\text{hot}}. \end{cases}$$

As it is represented on fig. 9, the negative sign inside $\Omega_{f,\text{hot}}$ ensures that $d_{\Omega_{f,\text{hot}}}$ is smooth on the boundary $\partial\Omega_{f,\text{hot}}$.

The non-mixing constraint of eq. (13) can equivalently be reformulated as stating that all points of the boundary $\partial\Omega_{f,\text{cold}}$ of the cold phase be at a distance at least $d_{\min}$ from the boundary of the hot domain:

$$\forall y \in \partial\Omega_{f,\text{cold}},\ d_{\Omega_{f,\text{hot}}}(y) \geq d_{\min}. \tag{14}$$

This formulation is still delicate to handle because it is a point-wise constraint which would translate in a numerical setting into as many constraints as there are vertices discretizing the boundary $\partial\Omega_{f,\text{cold}}$. A standard way to adress the difficulty is convenient

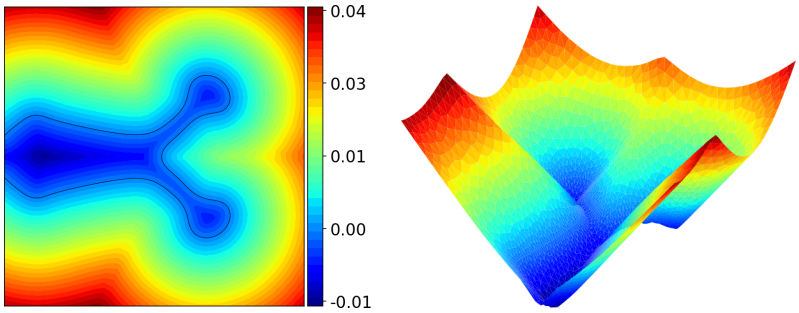


Figure 9: Isovalues of the signed distance function of a domain $\Omega_{f,\text{hot}} \subset D$. The boundary of $\Omega_{f,\text{hot}}$ is visible as the continuous black line within the square domain.

to approximate eq. (14) by a single averaged quantity. Since $d_{\Omega_{f,\text{hot}}}$ is a positive function on $\partial\Omega_{f,\text{cold}}$, eq. (14) can be formulated in terms of the supremum norm as

$$\left|\left|\frac{1}{d_{\Omega_{f,\text{hot}}}}\right|\right|^{-1}_{L^\infty(\partial\Omega_{f,\text{cold}})} \geq d_{\min}. \tag{15}$$

The infinity norm is then classically approximated with an L^p norm[1], for a sufficiently large value of p, in order to obtain a differentiable shape functional. A final possible approximation of eq. (13) reads then:

$$P_{\text{cold}\to\text{hot}}(\Omega_f) \geq d_{\min}, \tag{16}$$

where the averaged quantity $P_{\text{cold}\to\text{hot}}(\Omega_f)$ is defined by

$$P_{\text{cold}\to\text{hot}}(\Omega_f) := \left|\left|\frac{1}{d_{\Omega_{f,\text{hot}}}}\right|\right|^{-1}_{L^p(\partial\Omega_{f,\text{cold}})} = \left(\int_{\partial\Omega_{f,\text{cold}}} \frac{1}{|d_{\Omega_{f,\text{hot}}}|^p}\mathrm{d}s\right)^{-\frac{1}{p}}. \tag{17}$$

The quantity $P_{\text{cold}\to\text{hot}}(\Omega_f)$ can be interpreted as an averaged measure of the distance $d(\Omega_{f,\text{cold}}, \Omega_{f,\text{hot}})$ between both phases. From our own experience, it is sufficient to consider the value $p = 4$ to obtain a correct approximation of the constraint.

It remains to describe how the shape derivative of $P_{\text{cold}\to\text{hot}}$, which depends on $d_{\Omega_{f,\text{hot}}}$, can be computed. A basic application of shape calculus and of the chain rule yields

$$\mathrm{D}P_{\text{cold}\to\text{hot}}(\Omega_f)(\boldsymbol{\theta}) = -\frac{1}{p}P_{\text{cold}\to\text{hot}}(\Omega_f)^{p+1}\left[\int_{\partial\Omega_{f,\text{cold}}}\left(\frac{\partial}{\partial\boldsymbol{n}}+\kappa\right)\frac{1}{|d_{\Omega_{f,\text{hot}}}|^p}\boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}y - \int_{\partial\Omega_{f,\text{cold}}}\frac{p}{|d_{\Omega_{f,\text{hot}}}|^{p+1}}d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta})\mathrm{d}y,\right].$$

The first term of the above right-hand side has the classical structure of a boundary integral of the type of eq. (12) which can be easily numerically evaluated as a linear form in terms of $\boldsymbol{\theta}$. The second term involves the shape derivative $d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta})$ of the signed

[1]Note that we use the same notation for convenience but it is not to be confused pressure field also denoted by p.

distance function and is more delicate to evaluate. It is also possible to derive an explicit formula of the type eq. (12) for this term:

$$-\int_{\partial\Omega_{f,\text{cold}}} \frac{p}{|d_{\Omega_{f,\text{hot}}}|^{p+1}} d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta})\mathrm{d}y = \int_{\partial\Omega_{f,\text{hot}}} u(\Omega_{f,\text{hot}})\, \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}y, \tag{18}$$

however the analytical formula found for the integrand $u(\Omega_{f,\text{hot}})$ is a rather complicated expression which depends on integrals along the normal rays to the boundary of $\partial\Omega_{f,\text{hot}}$ and on its principal curvatures (see Allaire et al. (2016) as one of the first approaches relying on such explicit formulas).

In order to compute the shape derivative of $P_{\text{cold}\to\text{hot}}(\Omega_f)$ conveniently, we rely on our recent variational technique described in Feppon et al. (2020a) which enables to estimate the function $u(\Omega_{f,\text{hot}})$ numerically without resorting to its analytical expression. The function $u(\Omega_{f,\text{hot}}) \equiv u$ is obtained by solving the variational problem

Find $u \in V_\omega$ such that $\forall v \in V_\omega$,

$$\int_{\partial\Omega_{f,\text{hot}}} uv\mathrm{d}y + \int_D \omega(\nabla u\cdot\nabla d_{\Omega_{f,\text{hot}}})(\nabla v\cdot\nabla d_{\Omega_{f,\text{hot}}})\mathrm{d}x = \int_{\partial\Omega_{f,cold}} \frac{p}{|d_{\Omega_{f,\text{hot}}}|^{p+1}} v\mathrm{d}y, \tag{19}$$

where ω is a positive weight and V_ω a specific Sobolev space. The left hand-side is a positive symmetric bilinear form which can easily be discretized with the finite element method. The right-hand side has been obtained by simply substituting $d'_{\Omega_f,\text{hot}}(\boldsymbol{\theta})$ in eq. (18) by the opposite of an arbitrary test function $v \in V_\omega$. In order to solve eq. (19) numerically with piecewise linear finite elements, the weight ω should vanish on the *skeleton* of $\Omega_{f,\text{hot}}$ (the medial axis or equivalently the set where $d_{\Omega_{f,\text{hot}}}$ is not differentiable). There is obviously no unique choice of ω, one which leads to satisfying results in most of our applications is

$$\omega := \frac{1}{1+100|d_{\Omega_{f,\text{hot}}}\Delta d_{\Omega_{f,\text{hot}}}|^{3.5}}, \tag{20}$$

where we refer to Feppon et al. (2020a) for the numerical estimation of the Laplacian $\Delta d_{\Omega_{f,\text{hot}}}$.

The property given by eq. (18) follows by setting $v = d_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta})$ as a test function in eq. (19) and using the following characterization of $d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta})$ (see Delfour and Zolésio (1994); Bellettini (2013)),

$$\begin{cases} \nabla d_{\Omega_{f,\text{hot}}}\cdot\nabla d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta}) = 0 \text{ a.e. in } D \\ \qquad\qquad d'_{\Omega_{f,\text{hot}}}(\boldsymbol{\theta}) = -\boldsymbol{\theta}\cdot\boldsymbol{n} \text{ on } \partial\Omega_{f,\text{hot}} \end{cases} \text{for any } \boldsymbol{\theta}\in W^{1,\infty}(D,\mathbb{R}^d).$$

More generally, this method enables to compute shape derivatives of any functional depending on $d_{\Omega_{f,\text{hot}}}$ (or potentially on its gradient), allowing to consider a large variety of geometric constraints: minimum thickness, maximum thickness, overhang, etc...

4 Numerical topology optimization

In this last section, we describe how shape derivatives can be integrated in a whole numerical optimization procedure in order to solve the heat exchanger design problem of

eq. (6). This is achieved thanks to two main ingredients which are discussed in the next sections 4.1 and 4.2: (i) a first order non-linear constrained optimization algorithm for computing a *descent direction* based on the evaluation of shape derivatives and (ii) an algorithm for the iterative numerical updates of shapes. A last section 4.3 provides numerical illustrative examples of topologically optimized heat exchanger devices generated with this methodology.

4.1 Null space gradient flows for constrained shape optimization

There is a rather large literature proposing various constrained optimization algorithms, see e.g. Nocedal and Wright (1999); Svanberg (1987) or Dunning and Kim (2015) in the context of shape optimization based on the method of Hadamard. However, these algorithms are often not devised for the infinite dimensional context of shape optimization where the design variable is the shape of the interface Γ, or they require tedious tuning of algorithm metaparameters (such as the penalty parameters in the augmented Lagrangian method).

For these reasons, we propose here to rely on our *null space gradient flow* algorithm which is an extension of projected gradient flows to inequality constrained problems (see e.g. Yamashita (1980)), and which is very convenient and robust to use for level set based topology optimization. The next paragraphs provide an overview of the method exposed in full details in Feppon et al. (2019b). A user-friendly open-source python implementation of the optimization routine is available on gitlab:

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

4.1.1 Null space gradient flows

The null space optimization algorithm enables one to solve rather arbitrary equality and inequality constrained minimization problems of the form

$$\begin{aligned} &\min_{x\in\mathcal{X}} \quad J(x) \\ &s.t. \begin{cases} g_i(x) = 0 \text{ for } 1 \leq i \leq l \\ h_j(x) \leq 0 \text{ for } 1 \leq j \leq m, \end{cases} \end{aligned} \tag{21}$$

where $\mathcal{X}$ is a set equipped with some kind of *manifold* structure, J is the objective function and $(g_i)_{1\leq i\leq l}$ and $(h_j)_{1\leq j\leq m}$ are respectively the equality and inequality constraints.

The null space method finds local minimizer to eq. (21) by solving the following Ordinary Differential Equation (ODE) on the manifold $\mathcal{X}$:

$$\dot{x} = -\alpha_J \xi_J(x) - \alpha_C \xi_C(x) \tag{22}$$

where $-\xi_J(x)$ and $-\xi_C(x)$ are two orthogonal directions which are respectively called the *null space* steps and the *range space* steps. The *null space* step $-\xi_J(x)$ is the steepest descent direction for decreasing the objective function J which does not increase the violation of the constraints at first order. Its expression reads

$$\xi_J(x) := \nabla J(x) - \sum_{i=1}^{l} \lambda_i \nabla g_i(x) - \sum_{j=1}^{m} \mu_j \nabla h_j(x) \tag{23}$$

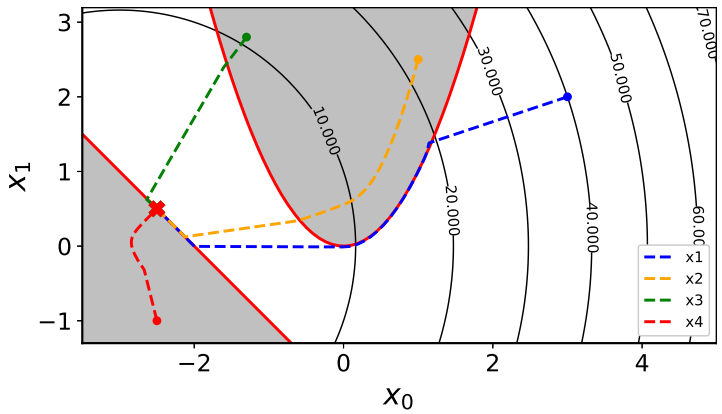
where $(\lambda_i)_{1\leq i\leq m}$ and $(\mu_j)_{1\leq j\leq m}$ are optimal Lagrange multipliers which can be obtained by solving a dual quadratic minimization problem. The *range space* step $-\xi_C(x)$ is a Gauss-Newton direction decreasing the violation of the constraints:

$$\xi_C(x) := \mathrm{D}C_{\widetilde{I}(x)}^T(\mathrm{D}C_{\widetilde{I}(x)}\mathrm{D}C_{\widetilde{I}(x)}^T)^{-1}C_{\widetilde{I}(x)}(x) \tag{24}$$

where $C_{\widetilde{I}(x)}(x) = \begin{bmatrix}(g_i(x))_{1\leq i\leq l} & (h_j(x))_{j\in\widetilde{I}(x)}\end{bmatrix}^T$ is the column vector gathering the equality constraints and the violated inequality constraints $\widetilde{I}(x) := \{1 \leq j \leq m \,|\, h_j(x) > 0\}$. Finally, α_J and α_C are two positive parameters which allows to control the pace at which the initial violation of the constraints needs to decrease.

The ODE of eq. (22) is a generalization of the standard gradient flow $\dot{x} = \nabla J(x)$ to constrained inequality and equality constrained problems. However, its discretization is not completely standard since its right-hand side can be discontinuous, leading to potential undesired oscillations. We refer the reader to Feppon et al. (2019b) for further information regarding its numerical treatment.

The continuous trajectories $x(t)$ solutions to eq. (22) have "good" mathematical properties: they gradually reach the feasible domain, before evolving tangentially to the constraints until finding a local minimizer satisfying the first order optimality (Karush Kuhn and Tucker) conditions. The typical behavior of these trajectories is illustrated on fig. 10 for a simple, but representative optimization problem. Optimization trajectories automat-



$$\begin{aligned} \min_{(x_1,x_2)\in\mathbb{R}^2} \quad & = x_1^2 + (x_2+3)^2 \\ \text{s.t.} \quad & \begin{cases} -x_1^2 + x_2 \leq 0 \\ -x_1 - x_2 - 2 \leq 0. \end{cases} \end{aligned} \tag{25}$$

Figure 10: Four optimization trajectories of the null space gradient flow for a simple minimization problem of eq. (25). The unfeasible domain is depicted in grey and the global minimizer is represented by the red cross.

ically detect whether they need to evolve tangentially to the inequality constraint barriers or if they rather need to go into the interior of the feasible domain. These properties are preserved after discretization of the ODE of eq. (22) so long as the time step selected for the time marching scheme is sufficiently small. Hence, the success of the optimization routine does not depend on the tuning of unphysical metaparameters.

4.1.2 Application of the null space algorithm to shape and topology optimization

In the context of solving the heat exchanger design problem of eq. (6), $\mathcal{X}$ is the set of shapes with sufficiently smooth (Lipschitz) boundaries:

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

while the time derivative $\dot{x}$ of eq. (22), which is the descent direction selected by the null space algorithm, is a deformation vector field $\dot{x} \equiv \boldsymbol{\theta} \in W^{1,\infty}(D, \mathbb{R}^d)$ for the current fluid-solid interface Γ. This vector field is computed from the knowledge of shape derivatives through the Riesz identification theorem which enables to convert linear forms into gradients. Considering a scalar-product $\langle\ ,\ \rangle_V$ on a Hilbert space $V \subset W^{1,\infty}(D, \mathbb{R}^3)$ of such vector fields, the gradient $\nabla F(\Omega_f) \in V$ associated to the shape derivative $\mathrm{D}F(\Omega_f)$ is the unique vector field satisfying

$$\forall \boldsymbol{\theta} \in V,\ \langle \nabla F(\Omega_f), \boldsymbol{\theta} \rangle_V = \mathrm{D}F(\Omega_f)(\boldsymbol{\theta}). \tag{26}$$

Typically, one considers $V = H^1(D, \mathbb{R}^d)$ and a common choice of the scalar product $\langle\ ,\ \rangle_V$ is

$$\langle \boldsymbol{\theta}, \boldsymbol{\theta}' \rangle_V := \int_D (\boldsymbol{\theta} \cdot \boldsymbol{\theta}' + h^2 \nabla \boldsymbol{\theta} : \nabla \boldsymbol{\theta}') \mathrm{d}x$$

where h is a regularization parameter homogeneous to a length and of the order of the mesh resolution. The identification operation of eq. (26) reduces to solve a linear system with right-hand side $\mathrm{D}F(\Omega_f)$; it takes place at every iteration of the null space optimization routine when computing the gradients of the objective and constraint functions in eq. (23), as well as the transpose $\mathrm{D}C^T_{\widetilde{I}(x)}$ of the differential of the violated constraints in eq. (24).

Finally, the discretization of the null space gradient flow given by eq. (22) with an Euler time step Δt and a deformation vector field $\boldsymbol{\theta}^n$ at iteration n requires to achieve an operation of the form

$$\Omega_f^{n+1} = \rho(\Omega_f^n, \Delta t \boldsymbol{\theta}^n) \tag{27}$$

where ρ is a mapping (called a "retraction") which updates the current shape Ω_f^n according to the descent direction $\Delta t \boldsymbol{\theta}^n$ into an improved design Ω_f^{n+1}. There is relative freedom for the step of eq. (27); we resort to a level set based remeshing approach which is explicited in the next section 4.2.

4.2 Body fitted mesh evolution method for numerical shape updates

We consider a body-fitted approach for the purpose of numerical shape updates based on the algorithm introduced in Allaire et al. (2014). This method allows complex topological change to occur in the course of the optimization while still maintaining an explicit discretization of the fluid and solid boundaries, which is especially relevant for fluid solvers needing to capture boundary layer effects in the vicinity of the fluid-solid interface. It brings the additional advantage of allowing users to use in principle their own physics solvers on the discretization mesh.

Let $\boldsymbol{\theta}^n$ be a given vector field which has been computed by the null space method on the nodes of a computational mesh $\mathcal{T}^n$ discretizing D and $\Omega_f \subset D$ (fig. 11.1). The

method provides as an output a new mesh $\mathcal{T}^{n+1}$ adapted to the discretization of the evolved domain $\Omega_f^{n+1} := \rho_{\Omega_f^n}(\Delta t \boldsymbol{\theta}^n)$. The algorithm, which implicitly defines the mapping ρ of the domain update as in eq. (27), consists in performing two key steps illustrated on fig. 12:

1. a deformation of Ω_f^n using the *level-set method* (Osher and Sethian (1988)) on the fixed mesh $\mathcal{T}^n$. The updated domain Ω_f^{n+1} is obtained implicitly on the mesh $\mathcal{T}^n$ as the negative subdomain of a level-set function ϕ^{n+1}:

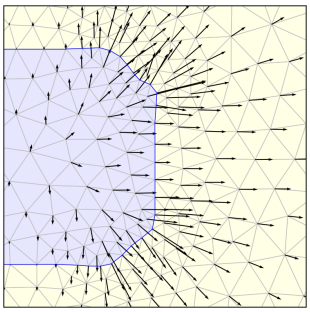
$$\Omega_f^{n+1} = \{x \in D \,|\, \phi^{n+1}(x) \leq 0\}.$$

   The function ϕ^{n+1} is computed by solving the *advection* equation

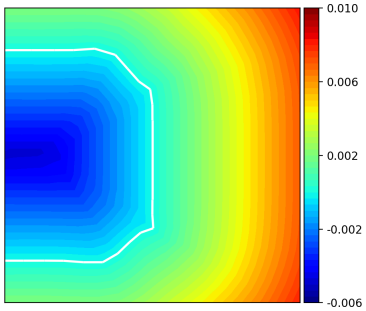
$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t,x) + \boldsymbol{\theta}(t,x) \cdot \nabla \phi(t,x) = 0, & x \in D, \\ \phi(0,x) = \phi^n(x), & x \in D, \end{cases}$$

   and by setting $\phi^{n+1}(x) := \phi(\Delta t, x)$. The initialization ϕ^n is a level-set function adapted to the current domain Ω_f^n (usually taken to be its signed distance function $d_{\Omega_f^n}$).

2. A *remeshing* step, which converts the current mesh $\mathcal{T}^n$ and the level-set ϕ^{n+1} into a new mesh $\mathcal{T}^{n+1}$ adapted to the new domain Ω_f^{n+1}. Such is achieved by splitting mesh elements of $\mathcal{T}$ according to the zero level set of ϕ^{n+1} before applying remeshing operations in order to obtain a high quality discretization mesh in view of numerical computations. In the numerical applications of our previous works, the remeshing operations are performed in 2D or in 3D thanks to the open-source software `mmg` (Dapogny et al. (2014); Cirrottola and Froehly (2019)).

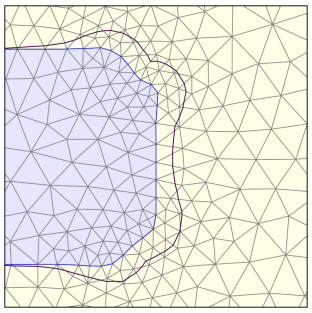


11.1: Vector field $\boldsymbol{\theta}^n$ on the mesh $\mathcal{T}^n$

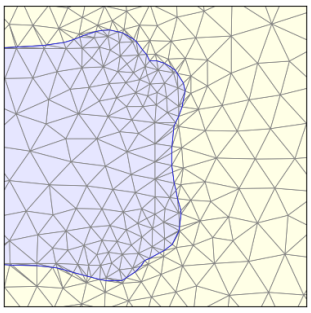


11.2: Level set function ϕ^n

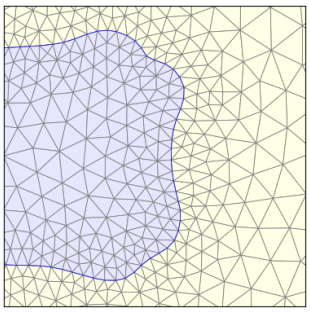
Figure 11: Illustration of the level set based mesh evolution algorithm of Allaire et al. (2013) on a 2D setting. A vector field $\boldsymbol{\theta}$ is defined at the vertices of the computational mesh $\mathcal{T}^n$ for the background domain D in which $\Omega_f^n \subset D$ is explicitly discretized (in blue on fig. 11.1). A level-set function ϕ^n adapted to Ω_f^n is computed on the mesh $\mathcal{T}^n$ (fig. 11.2). Figures from Feppon (2019); Feppon et al. (2019a).



12.1: Advection



12.2: Isosurface discretization



12.3: Remeshing

Figure 12: Mesh evolution algorithm illustrated on the 2D setting of fig. 11. 1. level set function ϕ^n is advected on $\mathcal{T}^n$ and yields a new level set function ϕ^{n+1} (fig. 12.1). 2. The mesh elements of $\mathcal{T}^n$ are split according to the zero level set of ϕ^{n+1} (fig. 12.2). 3. Remeshing operations are performed to obtain a new high quality mesh $\mathcal{T}^{n+1}$ adapted to the new domain Ω_f^{n+1} (fig. 12.3).

4.3 Numerical illustrations

We complete these notes with a few numerical illustrations of the method reported from our recent work Feppon et al. (2021). We consider the problem of designing a 3D heat exchanger featuring transverse cold and hot fluid channels in a cube $D = (0,1)^3$ as depicted on fig. 13. The input temperatures for the hot and cold channels are $T_{\text{hot}} = 100$ and $T_{\text{cold}} = 0$. The physical parameters of the state equations eqs. (1) and (3) are given by

$$\rho = 10,\ k_f = 1,\ k_s = 10,\ c_p = 200.$$

The design problem of eq. (6) supplemented with the volume constraints described by eq. (7) is solved with the following values for the maximum pressure drop, maximum volume and minimum wall thickness:

$$\texttt{DP}_0 = 3.72,\ \texttt{V}_0 = 0.15,\ d_{\min} = 0.04.$$

Note that for more robustness of the computational methodology, we implement the minimum distance constraint using two averaged distance constraints $P_{\text{cold}\to\text{hot}}(\Omega_f)$ and $P_{\text{hot}\to\text{cold}}(\Omega_f)$ rather than only the first one of them as in eq. (16). The results of the topology optimization methodology presented above are illustrated on the next figures which show the outcome of 500 iterations of the procedure. The state and adjoint equations of eqs. (1), (3), (10) and (11) where solved with parallel computing using the finite-element method on discretization meshes featuring up to 4 million of tetrahedra. Figure 14 shows the numerical evolution of the shapes of the total fluid domain Ω_f in the course of the iterations, starting from an initialization defined as two intricated array of pipes for each of the hot and cold fluid components. Complex topological changes occur in the course of the optimization process, but both fluid phases remain nonetheless well separated. The final design is shown in more details on fig. 15, where the final distribution of hot and cold fluid components are depicted as well as sectional views of the solid domain. Finally, the convergence histories of the objective and constraint functions are displayed on fig. 16. The behavior of the null space gradient flow algorithm of section 4.1 can be appreciated,

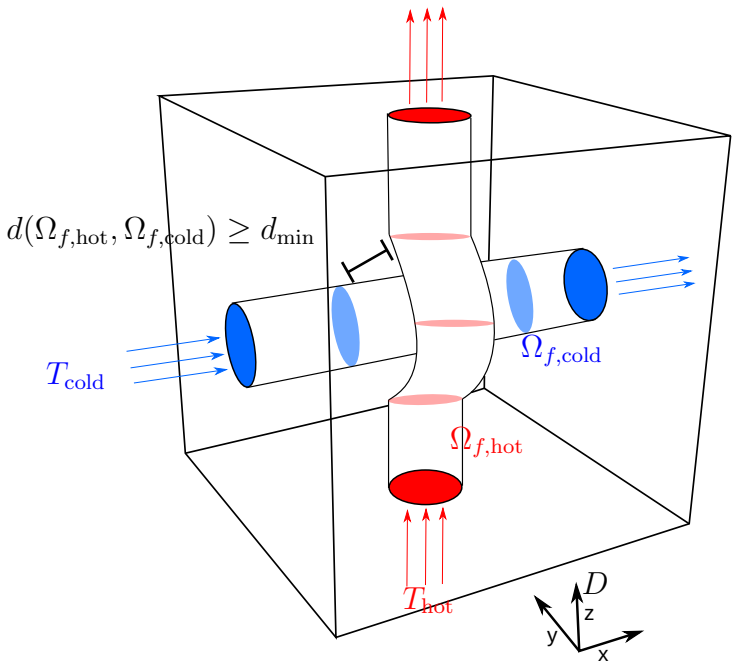


Figure 13: Schematic of the 3D heat exchanger design problem. Figure from Feppon et al. (2021).

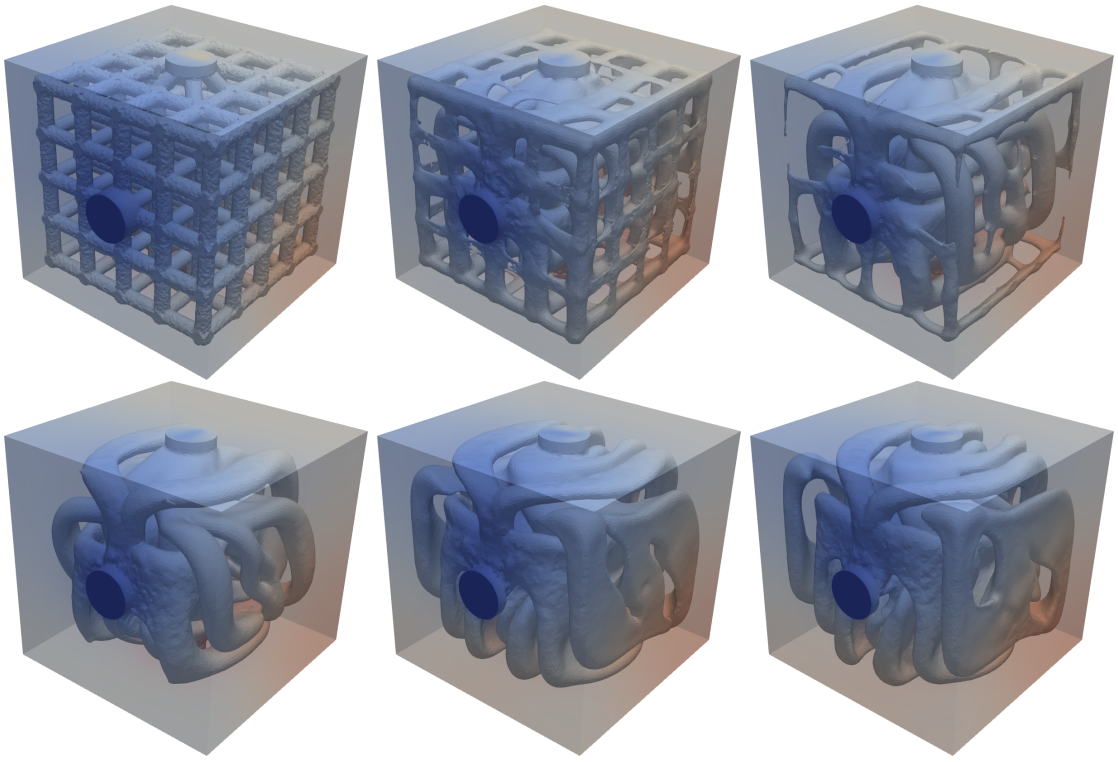
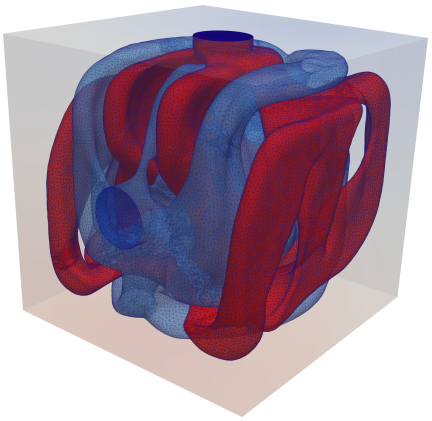
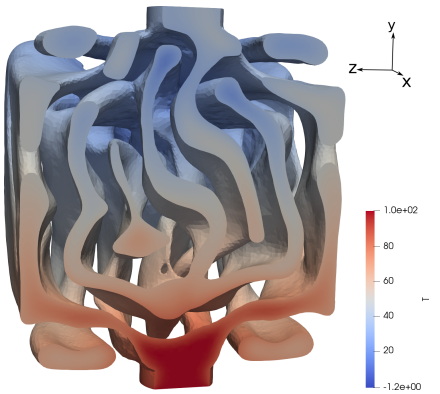


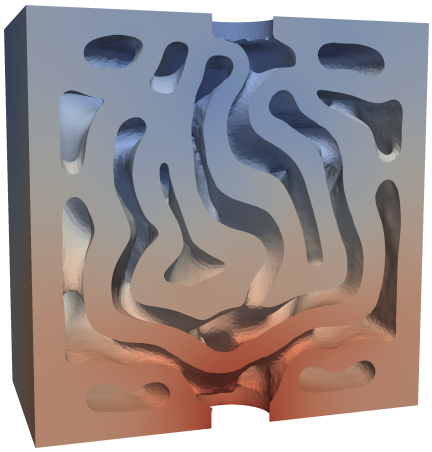
Figure 14: Iterations 0, 25, 50, 130, 200 and 500 of the topology optimization of the heat exchanger test case of fig. 13. The colors correspond to the magnitude of the temperature field. Figure reproduced from Feppon et al. (2021).



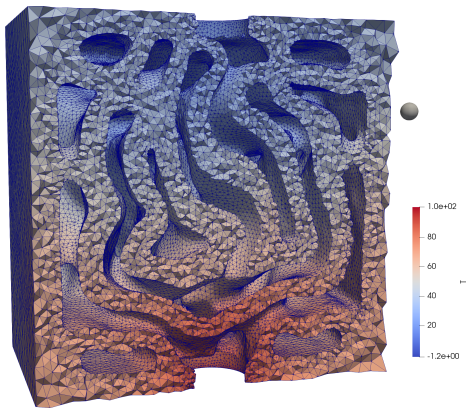
15.1: Optimized distribution of the cold and hot fluid components, respectively colored in blue and red.



15.2: Sectional view of the topologically optimized fluid domain. The color corresponds to the temperature profile.



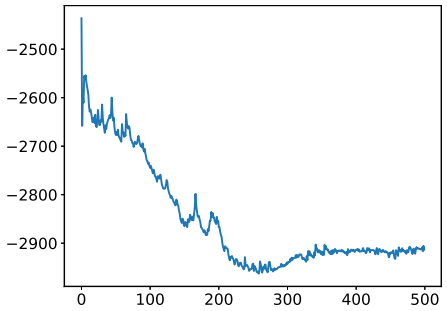
15.3: Sectional view of the topologically optimized solid domain.



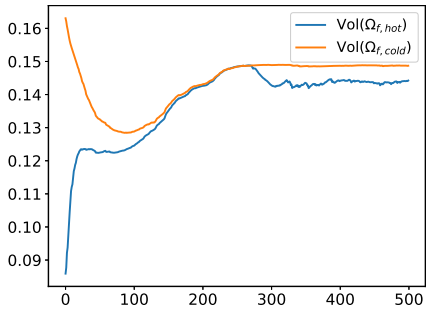
15.4: Sectional view of the topologically optimized solid domain with tetrahedral mesh elements made visible. The diameter of the sphere on the right-hand side is equal to the prescribed minimum distance constraint $d_{\min} = 0.04$.

Figure 15: Final design for the three-dimensional heat exchanger test case of fig. 13 with $d_{\min} = 0.04$. Figure reproduced from Feppon et al. (2021).

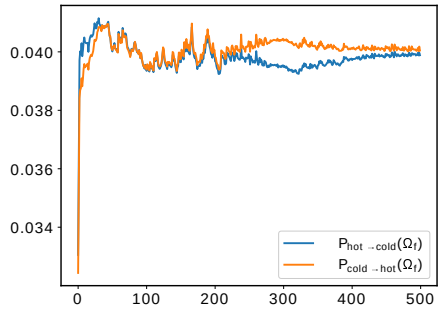
where one can verify that the objective function to minimize ($J = -\mathtt{W}$) keeps decreasing even after once the constraints are satisfied, which occur shortly after approximately 30 iterations. The method yields a design which is clearly more efficient than the initialization since the heat exchanged has increased by approximately 10%, but more importantly, it satisfies the maximum pressure drop and maximum volume constraints.



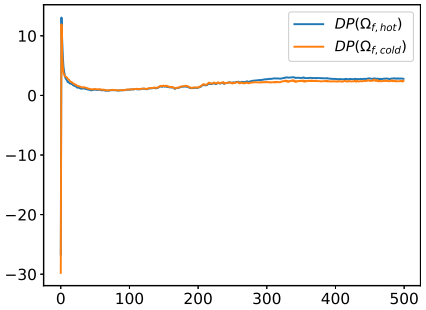
16.1: Objective function (opposite of the heat exchanged, $J = -\mathtt{W}$).



16.2: Volume fractions.



16.3: Averaged distance between the two phases.



16.4: Pressure drop.

Figure 16: Convergence histories of the 3D heat exchanger test case of fig. 13. Figure reproduced from Feppon et al. (2021).

5 Conclusions and perspectives

These notes provided an introductory exposure to a complete topology optimization methodology for automated design of multiple fluid phase heat exchangers. The cornerstones of the method are the computation of shape sensitivities, the need for taking into account a non-mixing constraint between the different input phases, an efficient constrained optimization algorithm and a level-set based mesh evolution method for the numerical shape updates. The efficiency of the approach has been illustrated on a moderately large-scale 3D case.

There is still room for progress towards a mature use of the method in industrial processes. Future works will certainly improve again the efficiency of the methodology by making use of parallel three-dimensional remeshing. Another improvement would be the extension of the methodology to more complex physical models involving turbulence or nonlinear thermal effects.

References

Allaire, G., Dapogny, C., and Frey, P. (2013). A mesh evolution algorithm based on the level set method for geometry and topology optimization. *Structural and Multidisciplinary Optimization*, 48(4):711–715.

Allaire, G., Dapogny, C., and Frey, P. (2014). Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282:22–53.

Allaire, G., Dapogny, C., and Jouve, F. (2021). Chapter 1 - shape and topology optimization. In Bonito, A. and Nochetto, R. H., editors, *Geometric Partial Differential Equations - Part II*, volume 22 of *Handbook of Numerical Analysis*, pages 1–132. Elsevier.

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

Allaire, G., Jouve, F., and Toader, A. (2002). A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334(12):1125–1130.

Awais, M. and Bhuiyan, A. A. (2018). Heat and mass transfer for compact heat exchanger (CHXs) design: A state-of-the-art review. *International Journal of Heat and Mass Transfer*, 127:359–380.

Bellettini, G. (2013). *Lecture notes on mean curvature flow, barriers and singular perturbations*, volume 12 of *Appunti. Scuola Normale Superiore di Pisa (Nuova Serie)*. Edizioni della Normale, Pisa.

Bhutta, M. M. A., Hayat, N., Bashir, M. H., Khan, A. R., Ahmad, K. N., and Khan, S. (2012). CFD applications in various heat exchangers design: A review. *Applied Thermal Engineering*, 32:1–12.

Capey, E. (1965). Alleviation of thermal stresses in aircraft structures. ministry of aviation, aeronautical research council. CP #819.

Cirrottola, L. and Froehly, A. (2019). Parallel unstructured mesh adaptation using iterative remeshing and repartitioning. Research Report 9307, INRIA Bordeaux, équipe CARDAMOM.

Dapogny, C., Dobrzynski, C., and Frey, P. (2014). Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378.

Delfour, M. and Zolésio, J. (1994). Shape analysis via distance functions. *J. Funct. Anal*, 123:129–201.

Delfour, M. C. and Zolesio, J.-P. (2011). *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, volume 22. Siam.

Dinh, Q. V., Rogé, G., Sevin, C., and Stoufflet, B. (1996). Shape optimization in computational fluid dynamics. *Revue Européenne des Éléments Finis. European Journal of Finite Elements*, 5:569–594.

Dunning, P. D. and Kim, H. A. (2015). Introducing the sequential linear programming level-set method for topology optimization. *Structural and Multidisciplinary Optimization*, 51(3):631–643.

Duta, M. C., Shahpar, S., and Giles, M. B. (2007). Turbomachinery design optimization using automatic differentiated adjoint code. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pages 1435–1444. American Society of Mechanical Engineers.

Feppon, F. (2019). *Shape and topology optimization of multiphysics systems*. PhD thesis, École Polytechnique.

Feppon, F., Allaire, G., Bordeu, F., Cortial, J., and Dapogny, C. (2019a). Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework. *SeMA*, 76(3):413–458.

Feppon, F., Allaire, G., and Dapogny, C. (2019b). Null space gradient flows for constrained optimization with applications to shape optimization. *ESAIM: COCV, Forthcoming article*.

Feppon, F., Allaire, G., and Dapogny, C. (2020a). A variational formulation for computing shape derivatives of geometric constraints along rays. *ESAIM: M2AN*, 54(1):181–228.

Feppon, F., Allaire, G., Dapogny, C., and Jolivet, P. (2020b). Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing. *Journal of Computational Physics*, 417:109574.

Feppon, F., Allaire, G., Dapogny, C., and Jolivet, P. (2021). Body-fitted topology optimization of 2d and 3d fluid-to-fluid heat exchangers. *Computer Methods in Applied Mechanics and Engineering*, 376:113638.

Giannoni, R. (1999). Gas-liquid heat exchanger and method for its manufacture. European Patent 1 098 156 A1.

Haertel, J. H. K. (2018). *Design of Thermal Systems Using Topology Optimization*. PhD thesis, Technical University of Denmark.

Harries, S., Abt, C., and Brenner, M. (2019). Upfront CAD—Parametric modeling techniques for shape optimization. In *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, volume 48, pages 191–211. Springer.

Henrot, A., Pierre, M., and Kawohl, B. (2018). *Shape Variation and Optimization, A Geometrical Analysis*, volume 121. Springer Fachmedien Wiesbaden GmbH.

Hiptmair, R., Paganini, A., and Sargheini, S. (2015). Comparison of approximate shape gradients. *BIT Numerical Mathematics*, 55(2):459–485.

Hoghoj, L. C., Norhave, D. R., Alexandersen, J., Sigmund, O., and Andreasen, C. S. (2020). Topology optimization of two fluid heat exchangers. *International Journal of Heat and Mass Transfer*, 163:120543.

Huang, G. P., Doman, D., Oppenheimer, M., Tipton, A., and Sigthorsson, D. (2019). Topology optimization of a fuel thermal management system. In *AIAA Aviation 2019 Forum*, page 3471.

Incropera, F. P., DeWitt, D. P., Bergman, T. L., and Lavine, A. S. (2012). *Foundations of Heat Transfer, 6th Edition International Student Version*. Wiley.

Kobayashi, H., Yaji, K., Yamasaki, S., and Fujita, K. (2020). Topology design of two-fluid heat exchange. *Structural and Multidisciplinary Optimization*, 63(2):821–834.

Laurent, M., Estevez, R., Fabrègue, D., and Ayax, E. (2016). Thermomechanical fatigue life prediction of 316L compact heat exchanger. *Engineering Failure Analysis*, 68:138–149.

Multiphysics, C. (1994). Comsol. *Inc., Burlington, MA, www. comsol. com*.

Nocedal, J. and Wright, S. J. (1999). Numerical optimization. *Springer Science*, 35(67-68).

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49.

Papazoglou, P. (2015). Topology optimization of heat exchangers. Master's thesis, TU Delft.

Pietropaoli, M., Montomoli, F., and Gaymann, A. (2019). Three-dimensional fluid topology optimization for heat transfer. *Structural and Multidisciplinary Optimization*, 59(3):801–812.

Rebei, A. (2019). *Développement de méthode d'optimisation topologique adaptée aux écoulements en régime turbulent, application au cas des échangeurs de chaleur*. PhD thesis, Université Paris Sciences et Lettres.

Saltzman, D., Bichnevicius, M., Lynch, S., Simpson, T. W., Reutzel, E. W., Dickman, C., and Martukanitz, R. (2018). Design and evaluation of an additively manufactured aircraft heat exchanger. *Applied Thermal Engineering*, 138:254–263.

Saviers, K. R., Ranjan, R., and Mahmoudi, R. (2019). Design and validation of topology optimized heat exchangers. In *AIAA Scitech 2019 Forum*, page 1465.

Shah, R. K. and Sekulic, D. P. (2003). *Fundamentals of heat exchanger design*. John Wiley & Sons.

Sokolowski, J. and Zolésio, J.-P. (1992). *Introduction to shape optimization*, volume 16 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin.

Svanberg, K. (1987). The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373.

Tawk, R., Ghannam, B., and Nemer, M. (2019). Topology optimization of heat and mass transfer problems in two fluids-one solid domains. *Numerical Heat Transfer, Part B: Fundamentals*, 76(3):130–151.

Yamashita, H. (1980). A differential equation approach to nonlinear programming. *Mathematical Programming*, 18(1):155–168.

Yu, M., Ruan, S., Gu, J., Ren, M., Li, Z., Wang, X., and Shen, C. (2020). Three-dimensional topology optimization of thermal-fluid-structural problems for cooling system design. *Structural and Multidisciplinary Optimization*, 62(6):3347–3366.