TOPOLOGY OPTIMIZATION OF THERMAL FLUID–STRUCTURE SYSTEMS USING BODY-FITTED MESHES AND PARALLEL COMPUTING

F. FEPPON[12], G. ALLAIRE[1], C. DAPOGNY[3], P. JOLIVET[4]

[1] Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France
[2] Safran Tech, Magny-les-Hameaux, France
[3] Univ. Grenoble Alpes, CNRS, Grenoble INP[1], LJK, 38000 Grenoble, France
[4] IRIT, CNRS, Toulouse, France

ABSTRACT. An efficient framework is described for the shape and topology optimization of realistic three-dimensional, weakly-coupled fluid-thermal-mechanical systems. At the theoretical level, the proposed methodology relies on the boundary variation of Hadamard for describing the sensitivity of functions with respect to the domain. From the numerical point of view, three key ingredients are used: (i) a level set based mesh evolution method allowing to describe large deformations of the shape while maintaining an adapted, high-quality mesh of the latter at every stage of the optimization process; (ii) an efficient constrained optimization algorithm which is very well adapted to the infinite-dimensional shape optimization context; (iii) efficient preconditioning techniques for the solution of large finite element systems in a reasonable computational time. The performance of our strategy is illustrated with two examples of coupled physics: respectively fluid–structure interaction and convective heat transfer. Before that, we perform three other test cases, involving a single physics (structural, thermal and aerodynamic design), for comparison purposes and for assessing our various tools: in particular, they prove the ability of the mesh evolution technique to capture very thin bodies or shells in 3D.

Keywords. Shape and topology optimization, fluid–structure interaction, convective heat transfer, aerodynamic design, mesh adaptation, distributed computing.

AMS Subject classifications. 74P10, 76B75, 74F10, 35Q79, 65N50, 65N55 .

CONTENTS

Corresponding author. Email: `florian.feppon@polytechnique.edu`.
[1]Institute of Engineering Univ. Grenoble Alpes

1. INTRODUCTION

Shape and topology optimization for multiphysic systems is still a huge challenge. Contrary to classical applications of optimal design techniques in aerodynamics, solely relying on geometric shape optimization [77, 54, 88], one of the difficulty is that here both the shape and topology have to be optimized, in order to obtain truly innovative designs. Furthermore, many applications, such as the design of aircraft engine components in the field of aeronautics, require to simultaneously consider fluid, thermal, and mechanical aspects in the optimization process. This raises new issues like the coupled discretization of the fluid and solid subdomains, or the computational efficiency of computing the system performance.

Fairly many works have proposed topology optimization methods for situations where several of the aforementioned physical effects arise: convective heat transfer problems (involving coupled fluid and thermal equations, where the elastic response of the underlying structure is neglected) have been addressed using density methods [73, 34, 38, 109, 97, 37, 36, 86, 91], or variants of the level-set method [4, 29, 107]. Systems featuring interactions between a fluid and a solid phase, without taking thermal effects into account, have been studied in slightly fewer works, and in two space dimensions only [108, 80, 14, 57, 72, 66]. Finally, the optimal design of thermoelastic structures has been tackled in [59, 100, 41, 106, 28, 10]. These problems are delicate, in part because of the very fine effects occurring at the interface between the fluid and solid phases at play. Furthermore, the numerical computation of coupled physics involving fluids in three dimensions is known to be very challenging.

The purpose of the present paper is to propose an efficient framework for three-dimensional shape and topology optimization of coupled fluid–solid thermal systems. The physical situations of interest are described by the incompressible Navier–Stokes equations in the fluid domain, the convection–diffusion equation for heat propagation in both fluid and solid domains, and the linearized elasticity system for the mechanical displacement of the solid domain.

Our method is based on our previous work [43] where we introduced the key ingredients of our framework in 2D: (i) analytic shape derivative formulas obtained with the method of Hadamard [81, 51, 101] and (ii) the level-set based mesh evolution algorithm of [7, 8, 9, 43] for the numerical realization of the deformation of meshed shapes. The main advantage of the latter strategy is that an explicit mesh discretization of the optimized fluid and solid domains remain available throughout the optimization process, while still allowing for dramatic deformations from one stage to the other (including topological changes), contrary to classical geometric methods. Also, in contrast with more commonly used density methods or with "classical" level-set methods [12, 104], this allows to solve the state equations without any modification or approximation of the original physics, e.g., without resorting to the Brinkman model for the fluid system, and without the need for considering relaxed "grayscale" designs, whose physical interpretation may be difficult, and which is prone to numerical inaccuracies, see [67, 65]. In particular, this approach is by nature non-intrusive: the physical state equations could in principle be solved by a "black box" external solver.

Another salient aspect of our methodology is the use of the null space optimization algorithm developed in one of our other previous works [44]. This algorithm is a generalization of gradient flow strategies for constrained optimization problems which is well-suited for the shape optimization applications of this paper. It allows us to solve a variety of optimization problems with a "small" number of quite arbitrary equality and inequality constraints, without requiring the tuning of unintuitive algorithmic parameters.

We are motivated to solve large-scale problems in this context, close to realistic industrial applications. If the validity of our overall multiphysics topology optimization algorithm was demonstrated numerically in our previous work [43] on two-dimensional test cases, its implementation in 3D is, classically, substantially more involved. Despite the fact that the theoretical ingredients of the topology optimization algorithm such as the expressions of shape derivatives remain identical, three-dimensional situations feature indeed two main sources of additional complexity.

On the one hand, the remeshing operations involved in our level-set based mesh evolution method are much more delicate and numerically expensive in three dimensions than in two dimensions. Our implementation extensively relies on the open-source platform Mmg [31] and its level-set discretization routines. The main features of the underlying algorithm are described in details in [30], with a particular emphasis on the three-dimensional case.

On the other hand, the physical state equations need to be solved about a hundred times during a typical shape optimization process which is very computationally demanding for large-scale problems, all the more when fluid mechanics is involved. The main difficulty to overcome when such applications are considered is the large size of the associated finite element problems. In this work, we resort to preconditioned iterative methods and to distributed computing techniques which allow to solve large-scale finite element problems in parallel at a reduced time- and memory-cost. This type of strategy is classical and has been implemented by many works relying on density methods in the context of the optimization of mechanical structures [2, 76, 13] or of heat convection systems [91, 86, 5, 23, 97]. Interestingly, this trend is also emerging in other design optimization methods, such as evolutionary topology optimization [75] or level-set methods [61]. As far as we are concerned, our numerical implementation relies on the PETSc library [17] and its interface in FreeFEM [49]. Let us emphasize that one originality of our work is the implementation of discretization and preconditioning techniques that are suited to isotropic or anisotropic unstructured meshes, which is a challenging numerical context with respect to the aforementioned works, where a fixed, or a structurally refined Cartesian grid is used.

The three-dimensional numerical examples tackled using our strategy make extensive use of parallel computing and preconditioning in order to be achieved in reasonable CPU time. Our largest test case features up to 5 million degrees of freedom solved on 24 processes. Admittedly, one may argue that their size remains rather small when compared to real-life industrial problems reaching about one billion degrees of freedom. However, we note that the algorithmic tools developed for this work, such as FreeFEM discretization or PETSc preconditioners, have the ability to scale on much larger architectures. Since ParMmg, the parallel counterpart of Mmg, does not handle yet adaptations of surfaces, we chose in the present article configurations that could be still tractable when using a sequential remesher. Nevertheless, we believe that these results are promising and pave the way to more challenging applications.

This article is organized as follows. In section 2, we describe our physical model and we review the main mathematical ingredients which enable to solve shape optimization problems with the method of Hadamard. In section 3, we briefly present the main ingredients of our numerical framework, namely the constrained optimization algorithm and the level-set based mesh evolution method used to deform the shape through the optimization process. Section 4 then discusses our choice of discretization and preconditioning techniques for the solution of the three-dimensional multiphysics state equations involved in our shape optimization test cases. Finally, section 5 is devoted to the presentation of a variety of three-dimensional test cases treated by our implementation. Five particular physical models are considered: while the first three of them deal with monophysics applications, in linear elasticity, heat conduction, or fluid mechanics, the fourth and fifth ones are true multiphysics examples; they arise in the respective contexts of fluid–structure interaction and convective heat transfer. We note that remeshing explicitly the fluid-solid interface at every iteration with a locally high resolution allowed us to obtain very original designs which could have been difficult to capture with more classical methods relying on fixed meshes.

2. Shape optimization of a multiphysics system with the method of Hadamard

This section sets the mathematical and physical frame of the article. We first describe in section 2.1 the three physics model coupling fluid, mechanical and thermal properties. We then present in section 2.2 a summary of the method of Hadamard for shape sensitivity analysis in this context.

2.1. Presentation of the three-physics model

Throughout this paper, $D = \Omega_s \cup \Omega_f$ is a fixed and bounded domain in $\mathbb{R}^d$, where $d = 2$ or 3 in practice. It is the disjoint reunion of a "fluid" phase Ω_f and a "solid" phase $\Omega_s = D \backslash \overline{\Omega_f}$, separated by an interface $\Gamma := \partial\Omega_f \cap \partial\Omega_s$, which we aim to optimize, see Figure 1.

Three coupled physical effects are at play within D:

(i) the motion of the fluid inside the region Ω_f, described by the velocity and pressure fields $\boldsymbol{v} : \Omega_f \to \mathbb{R}^d$ and $p : \Omega_f \to \mathbb{R}$;

(ii) the diffusion of heat inside the whole domain D and its transport by convection in the fluid domain, resulting in a temperature field $T : D \to \mathbb{R}$;
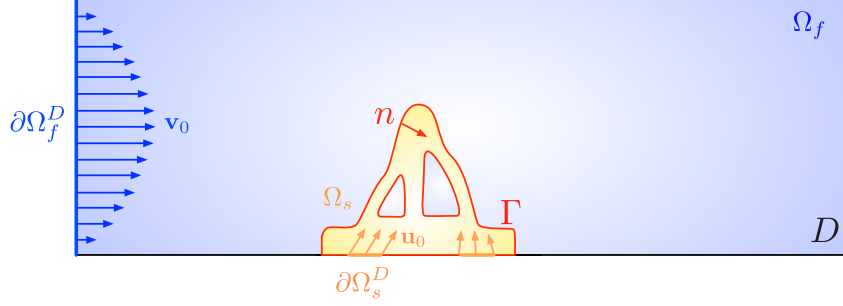
FIGURE 1. Setting of the considered three-physics problem, not all the regions of ∂D featured in (2.1) to (2.3) are represented.

(iii) the deformation of the solid region Ω_s, characterized by its mechanical displacement $\boldsymbol{u} : \Omega_s \to \mathbb{R}^d$, induced by both the stress applied by the fluid and the dilation under thermoelastic effects.

Let us now provide a little more details about the above models, referring to [43] and the references therein for the underlying mathematical theory.

The domain Ω_f contains a Newtonian fluid, with viscosity ν and density ρ. Its velocity $\boldsymbol{v}$ and pressure p satisfy the incompressible, steady-state Navier–Stokes equations:

$$
\begin{cases}
-\mathrm{div}(\sigma_f(\boldsymbol{v},p)) + \rho \nabla \boldsymbol{v}\, \boldsymbol{v} = \boldsymbol{f}_f & \text{in } \Omega_f \\
\mathrm{div}(\boldsymbol{v}) = 0 & \text{in } \Omega_f \\
\boldsymbol{v} = \boldsymbol{v}_0 & \text{on } \partial\Omega_f^D \\
\sigma_f(\boldsymbol{v},p)\boldsymbol{n} = 0 & \text{on } \partial\Omega_f^N \\
\boldsymbol{v} = 0 & \text{on } \Gamma.
\end{cases}
\tag{2.1}
$$

In the above system, $\nabla \boldsymbol{v}$ is the Jacobian matrix of $\boldsymbol{v}$, whose entries read $(\nabla \boldsymbol{v})_{ij} = \partial_j v_i$, $i,j = 1,\ldots,d$. The internal efforts within the fluid are described by the fluid stress tensor $\sigma_f(\boldsymbol{v},p)$ which is related to the rate of strain tensor $e(\boldsymbol{v}) := \frac{1}{2}(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)$ via the Newton's law:

$$
\sigma_f(\boldsymbol{v},p) := 2\nu e(\boldsymbol{v}) - p\mathrm{I},
$$

where I is the $d \times d$ identity matrix; the external stresses, such as gravity, are modeled by a body force $\boldsymbol{f}_f$. The system (2.1) features a decomposition of the boundary of the fluid phase as the disjoint reunion $\partial\Omega_f = \partial\Omega_f^D \cup \partial\Omega_f^N \cup \Gamma$ of

(i) a Dirichlet (or inlet) part $\partial\Omega_f^D$, where the flow enters with given velocity $\boldsymbol{v} = \boldsymbol{v}_0$;
(ii) a Neumann (or outlet) part $\partial\Omega_f^N$, where zero normal stress is observed;
(iii) the interface Γ with the solid domain Ω_s, where no-slip boundary conditions are imposed.

The unit normal vector $\boldsymbol{n} : \Gamma \to \mathbb{R}^d$ is oriented from Ω_f to Ω_s.

From the knowledge of the velocity of the fluid $\boldsymbol{v}$, it is possible to identify the temperature T in the total domain D, as a result of convection and diffusion inside the fluid domain Ω_f, and of pure diffusion inside the solid domain Ω_s. Denoting by k_f and k_s the thermal conductivity coefficients inside Ω_f and Ω_s respectively, and by c_p the thermal capacity of the fluid, the temperature T is the solution to the steady-state

convection–diffusion equations:

$$\begin{cases} -\mathrm{div}(k_f \nabla T_f) + \rho c_p \boldsymbol{v} \cdot \nabla T_f = Q_f & \text{in } \Omega_f \\ -\mathrm{div}(k_s \nabla T_s) = Q_s & \text{in } \Omega_s \\ T = T_0 & \text{on } \partial\Omega_T^D \\ -k_f \dfrac{\partial T_f}{\partial \boldsymbol{n}} = h & \text{on } \partial\Omega_T^N \cap \partial\Omega_f \\ -k_s \dfrac{\partial T_s}{\partial \boldsymbol{n}} = h & \text{on } \partial\Omega_T^N \cap \partial\Omega_s \\ T_f = T_s & \text{on } \Gamma \\ -k_f \dfrac{\partial T_f}{\partial \boldsymbol{n}} = -k_s \dfrac{\partial T_s}{\partial \boldsymbol{n}} & \text{on } \Gamma, \end{cases} \tag{2.2}$$

where the subscripts $_f$ and $_s$ denote restrictions T_f and T_s of T to Ω_f and Ω_s respectively. The temperature T as well as the normal heat flux are continuous across the interface Γ between Ω_f and Ω_s, and volumic sources Q_f and Q_s are at work inside Ω_f and Ω_s. This system features a decomposition of the boundary $\partial D = \partial\Omega_T^N \cup \partial\Omega_T^D$ into a Dirichlet part $\partial\Omega_T^D$ where a temperature T_0 is imposed, and a Neumann part $\partial\Omega_T^N$ where a given incoming heat flux h is applied. Note that the regions $\partial\Omega_T^N$ and $\partial\Omega_T^D$ are independent of those $\partial\Omega_f^D$, $\partial\Omega_f^N$, and Γ featured in the system (2.1). For instance, $\partial\Omega_T^N$ may partially or totally overlap $\partial\Omega_f^N$ or $\partial\Omega_f^D$.

Eventually, the velocity $\boldsymbol{v}$ and pressure p inside the fluid domain Ω_f, together with the temperature T inside the total domain D, allow to determine the displacement $\boldsymbol{u}$ of the solid domain Ω_s. The latter is filled with an isotropic thermoelastic material with Lamé coefficients λ and μ, thermal expansion parameter α, and temperature at rest T_{ref}, so that $\boldsymbol{u}$ is the solution of the linear thermoelasticity system:

$$\begin{cases} -\mathrm{div}(\sigma_s(\boldsymbol{u}, T_s)) = \boldsymbol{f}_s & \text{in } \Omega_s \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{on } \partial\Omega_s^D \\ \sigma_s(\boldsymbol{u}, T_s)\boldsymbol{n} = \boldsymbol{g} & \text{on } \partial\Omega_s^N \\ \sigma_s(\boldsymbol{u}, T_s)\boldsymbol{n} = \sigma_f(\boldsymbol{v}, p)\boldsymbol{n} & \text{on } \Gamma, \end{cases} \tag{2.3}$$

where the solid stress tensor is given by

$$\sigma_s(\boldsymbol{u}, T_s) := Ae(\boldsymbol{u}) - \alpha(T_s - T_{\mathrm{ref}})\mathrm{I} \quad \text{with} \quad Ae(\boldsymbol{u}) := 2\mu e(\boldsymbol{u}) + \lambda \mathrm{Tr}(e(\boldsymbol{u}))\mathrm{I}, \tag{2.4}$$

and $\boldsymbol{f}_s$ is an applied body force. In (2.3), the boundary $\partial\Omega_s$ is decomposed as $\partial\Omega_s = \partial\Omega_s^D \cup \partial\Omega_s^N \cup \Gamma$, where:

(i) the displacement $\boldsymbol{u} = \boldsymbol{u}_0$ is prescribed on the Dirichlet part $\partial\Omega_s^D$;

(ii) a stress $\boldsymbol{g}$ is imposed on the Neumann part $\partial\Omega_s^N$;

(iii) the solid is subject to the pressure imposed by the fluid on the interface Γ with Ω_f, i.e., equality holds between the normal components of the fluid and solid stress tensors: $\sigma_f(\boldsymbol{v}, p)\boldsymbol{n} = \sigma_s(\boldsymbol{u}, T_s)\boldsymbol{n}$.

Again, this decomposition of $\partial\Omega_s$ is independent of the decompositions of $\partial\Omega_f$ and ∂D featured in the systems (2.1) and (2.2).

Note that the coupling between the above three physical systems is weak, insofar as the deformation of the fluid domain is neglected: as a result there is no influence of the solid deformation $\boldsymbol{u}$ onto the fluid variables $(\boldsymbol{v}, p)$, see [43] for related discussions.

2.2. Shape sensitivities of three-physics systems using the method of Hadamard

In the physical situation of section 2.1, we consider design optimization problems of the generic form:

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

$$\text{s.t.} \quad \begin{cases} g_i\left(\Gamma, \boldsymbol{v}(\Gamma), l(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma)\right) = 0, & 1 \leqslant i \leqslant p, \\ h_j\left(\Gamma, \boldsymbol{v}(\Gamma), m(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma)\right) \leqslant 0, & 1 \leqslant j \leqslant q, \end{cases} \tag{2.5}$$

where J is the objective function measuring the performance of the system and g_i (resp. h_j) are l equality (resp. m inequality) constraints. These functions all depend on the subdivision of the working domain D

into the fluid and solid parts Ω_f, Ω_s, either explicitly, or via the physical variables $\boldsymbol{v}(\Gamma), p(\Gamma), \boldsymbol{u}(\Gamma)$, and $T(\Gamma)$ determined by (2.1) to (2.3). For notational simplicity, and with a little abuse, this dependence is made explicit with the sole mention to Γ instead of referring to Ω_f and Ω_s. Several examples of such objective and constraint functionals are provided in the applicative section 5.

Like most first-order optimization strategies, e.g., gradient descent methods, our numerical framework for solving (2.5) rests on the derivative of the objective function J and constraints g_i, h_j with respect to the position of the fluid–solid interface Γ. This notion can be understood in a variety of manners. In the present work, we rely on the method of Hadamard [50, 6, 101], which considers variations of Γ of the form

$$\Gamma_{\boldsymbol{\theta}} := (\mathrm{Id} + \boldsymbol{\theta})(\Gamma), \text{ where } \boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \ ||\boldsymbol{\theta}||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1, \tag{2.6}$$

where $\boldsymbol{\theta}$ is a small vector field deforming Γ, see Figure 2. The shape derivative of a function $\Gamma \mapsto F(\Gamma)$ of the two-phase decomposition $D = \Omega_f \cup \Omega_s$ is then defined as the derivative $\mathrm{D}F(\Gamma) : W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) \to \mathbb{R}$ of the underlying mapping $\boldsymbol{\theta} \mapsto F(\Gamma_{\boldsymbol{\theta}})$ at $\boldsymbol{\theta} = 0$, so that the following asymptotic expansion holds:

$$F(\Gamma_{\boldsymbol{\theta}}) = F(\Gamma) + \mathrm{D}F(\Gamma)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \text{ where } \frac{|o(\boldsymbol{\theta})|}{||\boldsymbol{\theta}||_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)}} \xrightarrow{\boldsymbol{\theta} \to 0} 0. \tag{2.7}$$

In principle, one additionnally imposes that the vector fields $\boldsymbol{\theta}$ in the above definition vanish on the boundary ∂D of the total domain D, so that it is unchanged upon deformation by $\boldsymbol{\theta}$.
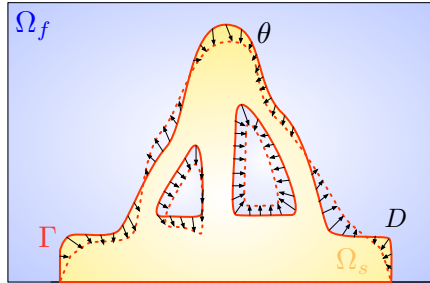

FIGURE 2. Deformation of the interface Γ by a small vector field $\boldsymbol{\theta}$ in the context of the boundary variation method of Hadamard.

The objective and constraint functionals considered in our applications have the general structure

$$\Gamma \mapsto J(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma)), \tag{2.8}$$

and so their dependence on the solid-fluid interface Γ, via the solution of the partial differential equations (2.1) to (2.3) posed on the corresponding phases Ω_f and Ω_s, is not trivial. Nevertheless, analytic expressions for these derivatives have been computed for quite arbitrary shape functionals J in [43]. They can be assembled automatically from the sole knowledge of the partial derivatives of J thanks to the solution of adequate adjoint systems. Since this is the most essential ingredient of our methodology, we reproduce this result below, referring to [43], Proposition 4, for the proof and related comments.

Proposition 1. *Let $\mathfrak{J}$ be the transported objective function defined by*

$$\forall \boldsymbol{\theta} \in W^{1,\infty}(D, \mathbb{R}^d) \text{ with } \boldsymbol{\theta} = 0 \text{ on } \partial D, (\widehat{\boldsymbol{v}}, \widehat{p}, \widehat{T}, \widehat{\boldsymbol{u}}) \in H^1(\Omega_f, \mathbb{R}^d) \times (L^2(\Omega_f)/\mathbb{R}) \times H^1(D) \times H^1(\Omega_s, \mathbb{R}^d),$$

$$\mathfrak{J}(\boldsymbol{\theta}, \widehat{\boldsymbol{v}}, \widehat{p}, \widehat{T}, \widehat{\boldsymbol{u}}) := J(\Gamma_{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}} \circ (I + \boldsymbol{\theta})^{-1}, \widehat{p} \circ (I + \boldsymbol{\theta})^{-1}, \widehat{T} \circ (I + \boldsymbol{\theta})^{-1}, \widehat{\boldsymbol{u}} \circ (I + \boldsymbol{\theta})^{-1}). \tag{2.9}$$

If $\mathfrak{J}$ has continuous partial derivatives then the objective function J in (2.8) is shape differentiable and the derivative reads

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

$$= \frac{\partial\mathfrak{J}}{\partial\boldsymbol{\theta}}(\boldsymbol{\theta}) + \int_{\Omega_f}[\boldsymbol{w}\cdot\mathrm{div}(\boldsymbol{f}_f\otimes\boldsymbol{\theta}) - (\sigma_f(\boldsymbol{v},p):\nabla\boldsymbol{w} + \rho\boldsymbol{w}\cdot\nabla\boldsymbol{v}\cdot\boldsymbol{v})\mathrm{div}(\boldsymbol{\theta})]\,\mathrm{d}x$$

$$+ \int_{\Omega_f}[\sigma_f(\boldsymbol{v},p):(\nabla\boldsymbol{w}\nabla\boldsymbol{\theta}) + \sigma_f(\boldsymbol{w},q):(\nabla\boldsymbol{v}\nabla\boldsymbol{\theta}) + \rho\boldsymbol{w}\cdot(\nabla\boldsymbol{v}\nabla\boldsymbol{\theta})\cdot\boldsymbol{v}]\mathrm{d}x$$

$$- \int_{\Omega_s}\mathrm{div}(\boldsymbol{\theta})(k_s\nabla T\cdot\nabla S)\mathrm{d}x - \int_{\Omega_f}\mathrm{div}(\boldsymbol{\theta})(k_f\nabla T\cdot\nabla S + \rho c_p(\boldsymbol{v}\cdot\nabla T)S)\mathrm{d}x$$

$$+ \int_{\Omega_s}k_s(\nabla\boldsymbol{\theta}+\nabla\boldsymbol{\theta}^T)\nabla T\cdot\nabla S\mathrm{d}x + \int_{\Omega_f}\left[k_f(\nabla\boldsymbol{\theta}+\nabla\boldsymbol{\theta}^T)\nabla T\cdot\nabla S + \rho c_p\boldsymbol{v}\cdot(\nabla\boldsymbol{\theta}^T\nabla T)S\right]\mathrm{d}x$$

$$+ \int_{\Omega_s}\mathrm{div}(Q_s\boldsymbol{\theta})S\mathrm{d}x + \int_{\Omega_f}\mathrm{div}(Q_f\boldsymbol{\theta})S\mathrm{d}x$$

$$+ \int_{\Omega_s}[-\mathrm{div}(\boldsymbol{\theta})\sigma_s(\boldsymbol{u},T):\nabla\boldsymbol{r} + \sigma_s(\boldsymbol{u},T):(\nabla\boldsymbol{r}\nabla\boldsymbol{\theta}) + Ae(\boldsymbol{r}):(\nabla\boldsymbol{u}\nabla\boldsymbol{\theta}) + \boldsymbol{r}\cdot\mathrm{div}(\boldsymbol{f}_s\otimes\boldsymbol{\theta})]\,\mathrm{d}x,\quad(2.10)$$

where $\boldsymbol{r},S,\boldsymbol{w}$, and q are the adjoint state solutions of the following variational problems:

$$\int_{\Omega_s}Ae(\boldsymbol{r}):\nabla\boldsymbol{r}'\mathrm{d}x = \frac{\partial\mathfrak{J}}{\partial\widehat{\boldsymbol{u}}}(\boldsymbol{r}')\quad\forall\boldsymbol{r}'\in V_{\boldsymbol{u}}(\Gamma)\,,\qquad(2.11)$$

$$\int_{\Omega_s}k_s\nabla S\cdot\nabla S'\mathrm{d}x + \int_{\Omega_f}(k_f\nabla S\cdot\nabla S' + \rho c_p S\boldsymbol{v}\cdot\nabla S')\mathrm{d}x = \int_{\Omega_s}\alpha\mathrm{div}(\boldsymbol{r})S'\mathrm{d}x + \frac{\partial\mathfrak{J}}{\partial\widehat{T}}(S)\quad\forall S'\in V_T(\Gamma)\,,\,(2.12)$$

$$\boldsymbol{w} = \boldsymbol{r}\ \text{on}\ \Gamma\ \text{and}\ \forall(\boldsymbol{w}',q')\in V_{\boldsymbol{v},p}(\Gamma),$$

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

$$\int_{\Omega_f}-\rho c_p S\nabla T\cdot\boldsymbol{w}'\mathrm{d}x + \frac{\partial\mathfrak{J}}{\partial(\boldsymbol{v}',p')}(\boldsymbol{w}',q'),\quad(2.13)$$

and where $V_{\boldsymbol{u}}(\Gamma)$, $V_T(\Gamma)$, and $V_{\boldsymbol{v},p}(\Gamma)$ are the functional spaces:

$$V_{\boldsymbol{u}}(\Gamma) := \{\boldsymbol{r}\in H^1(\Omega_s,\mathbb{R}^d)\,|\,\boldsymbol{r} := 0\ \text{on}\ \partial\Omega_s^D\},\qquad(2.14)$$

$$V_T(\Gamma) := \{S\in H^1(D)\,|\,S := 0\ \text{on}\ \partial\Omega_T^D\},\qquad(2.15)$$

$$V_{\boldsymbol{v},p}(\Gamma) := \{(\boldsymbol{w},q)\in H^1(\Omega_f,\mathbb{R}^d)\times L^2(\Omega_f)/\mathbb{R}\,|\,\boldsymbol{w}=0\ \text{on}\ \partial\Omega_f\}.\qquad(2.16)$$

Remark 1. Usually, (2.10) is called the volume form of the shape derivative because it involves volume integrals on Ω_s and Ω_f. Under technical regularity assumptions, this derivative can be rewritten in terms of surface integrals over the optimized boundary Γ, see Proposition 5 in [43] in the present setting. Although mathematically equivalent, these two expressions yield different implementations, with different numerical behaviors, see [42] for some numerical comparisons in the present context and [53, 110, 70] for further developments. In the present article, all our examples relied on the volume expression (2.10) of the shape derivative, except the one of section 5.3 where we used the surface expression in proposition 2 below.

3. Numerical resolution of the shape optimization problem

In this section, we provide a few details about the numerical methods involved in our treatment of the shape optimization problem (2.5). More precisely, in section 3.1, we briefly present our optimization algorithm for finding a suitable "descent" direction $\boldsymbol{\theta}$ with respect to the problem (2.5). Then, in section 3.2, we recall the main features of the level-set based mesh evolution method allowing to update numerically the shape according to this deformation. The solution of discrete finite element systems is deliberately left aside from this section, this important point being the central topic of section 4.

3.1. Null space gradient flow optimization algorithm

The constrained shape optimization problem (2.5) is solved thanks to the null space algorithm developed in [44]. The latter calculates a deformation $\boldsymbol{\theta}$ of a given shape Γ, such that the value of the objective function $J(\Gamma)$ is decreased, while ensuring that the constraints g_i and h_j are "better satisfied".

For simplicity, let us first briefly expose the method in a finite dimensional counterpart of (2.5), namely

$$\min_{x \in \mathbb{R}^n} \quad J(x)$$
$$\text{s.t.} \quad \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leqslant 0, \end{cases} \tag{3.1}$$

with l and m equality and inequality constraint vectors $\boldsymbol{g}(x) \in \mathbb{R}^l$ and $\boldsymbol{h}(x) \in \mathbb{R}^m$. Our method finds a local optimum of (3.1) by solving the following "null space gradient flow" in pseudo-time $t > 0$:

$$\dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)). \tag{3.2}$$

The ordinary differential equation (3.2) features two orthogonal directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$, defined by:

$$\boldsymbol{\xi}_J(x) := (\mathrm{I} - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^T (\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^T)^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}) \nabla J(x), \tag{3.3}$$

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T (\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)} \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T)^{-1} \boldsymbol{C}_{\widetilde{I}(x)}(x), \tag{3.4}$$

where for a given subset of indices $I \subset \{1, \dots, q\}$, the notation $\boldsymbol{C}_I(x) := \begin{bmatrix} \boldsymbol{g}(x) & \boldsymbol{h}_I(x) \end{bmatrix}^T$ stands for the vector gathering $\boldsymbol{g}$ and the components of $\boldsymbol{h}$ indexed by I. The subset $\widetilde{I}(x)$ featured in (3.4) is that of the saturated or violated inequality constraints (we denote by $\widetilde{m}(x)$ its number of elements):

$$\widetilde{I}(x) := \{i \in \{1, \dots, m\} \mid h_i(x) \geqslant 0\} \text{ with } \widetilde{m}(x) := \mathrm{Card}(\widetilde{I}(x)).$$

In a nutshell, $\boldsymbol{\xi}_C(x)$ in (3.4) is a Gauss–Newton direction aiming at decreasing the violation of the constraints, as measured by $\boldsymbol{C}_{\widetilde{I}(x)}(x)$. The "null space" direction $\boldsymbol{\xi}_J(x)$ is the orthogonal projection of the gradient $\nabla J(x)$ tangentially to a subset $\widehat{I}(x) \subset \widetilde{I}(x)$ of constraints which must remain active. This latter subset is optimal insofar as it contains exactly the indices of those saturated or violated inequality constraints that must be enforced at the current stage; the other inequality constraints h_j, for $j \notin \widehat{I}(x)$, naturally end up "better satisfied" when moving along the direction $-\boldsymbol{\xi}_J(x)$. Mathematically, $\widehat{I}(x)$ is characterized by the solution to the dual quadratic subproblem

$$(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) := \arg \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^l \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{m}(x)}, \, \boldsymbol{\mu} \geqslant 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^T \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^T \boldsymbol{\mu}||, \tag{3.5}$$

and by setting $\widehat{I}(x) := \{i \in \widetilde{I}(x) \mid \boldsymbol{\mu}_i^*(x) > 0\}$. Hence, $-\boldsymbol{\xi}_J(x)$ is the best possible descent direction respecting locally both equality and inequality constraints.

Finally, the optimization problem (3.1) is solved by using a Euler discretization scheme with time step Δt:

$$x_{n+1} = x_n - \Delta t (\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)). \tag{3.6}$$

The main practical advantage of the method lies in the fact that it does not require the tuning of non-physical parameters. It is enough that the time step Δt be sufficiently small, while the parameters α_J and α_C are easily set, depending on how fast the initially violated constraints must be enforced.

The above strategy can be adapted to deal with constrained shape optimization problems of the form (2.5), where $\mathbb{R}^n$ is replaced by a set of shapes Γ, given the following adjustments.

1. The definitions of the null space and Gauss–Newton directions $\boldsymbol{\xi}_J(\Gamma)$ and $\boldsymbol{\xi}_C(\Gamma)$ now involve shape derivatives:

$$\boldsymbol{\xi}_J(\Gamma) := (\mathrm{I} - \mathrm{D}\boldsymbol{C}_{\widehat{I}(\Gamma)}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C}_{\widehat{I}(\Gamma)} \mathrm{D}\boldsymbol{C}_{\widehat{I}(\Gamma)}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(\Gamma)}) \nabla J(\Gamma), \tag{3.7}$$

$$\boldsymbol{\xi}_C(\Gamma) := \mathrm{D}\boldsymbol{C}_{\widetilde{I}(\Gamma)}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C}_{\widetilde{I}(\Gamma)} \mathrm{D}\boldsymbol{C}_{\widetilde{I}(\Gamma)}^{\mathcal{T}})^{-1} \boldsymbol{C}_{\widetilde{I}(\Gamma)}(\Gamma), \tag{3.8}$$

where the gradient $\nabla J(\Gamma)$ and the transpose shape derivative $\mathrm{D}\boldsymbol{C}_{\widehat{I}(\Gamma)}^{\mathcal{T}}$ are computed by taking advantage of the classical regularization–extension operation of shape derivatives, see [25, 24, 16, 15, 35] and section 5 of [43]. For instance, in our case, $\nabla J(\Gamma)$ is obtained by solving the following Riesz identification elliptic problem:

$$\text{find } \nabla J(\Gamma) \in H^1(D, \mathbb{R}^d) \text{ s.t. } \forall \boldsymbol{\theta} \in H^1(D, \mathbb{R}^d), \int_D (\gamma^2 \nabla J(\Gamma) : \nabla \boldsymbol{\theta} + \nabla J(\Gamma) \cdot \boldsymbol{\theta}) \mathrm{d}x = \mathrm{D}J(\Gamma)(\boldsymbol{\theta}), \qquad (3.9)$$

where γ is a regularization length scale typically set to a few times the mesh element size. In a similar spirit, the transpose $\mathcal{T}$ of (3.7) and (3.8) is different from the usual matrix transpose T in (3.3) and (3.4), see section 2.1 in [44] for the mathematical definition.

2. According to the method of Hadamard, the Euler scheme (3.6) is replaced with the following design update step:

$$\Gamma_{n+1} = (\mathrm{Id} - \Delta t(\alpha_J \boldsymbol{\xi}_J(\Gamma_n) + \alpha_C \boldsymbol{\xi}_C(\Gamma_n)))(\Gamma_n).$$

3.2. **Level-set based mesh evolution method**

When it comes to the numerical representation of shapes and their deformations in the course of the optimization process, we rely on the level-set based mesh evolution strategy introduced in [9], see also [85] for a preliminary version of this methodology. The main idea of this method is to take advatdge of two numerical representations of the fluid subdomain $\Omega_f \subset D$, or equivalently the solid subdomain Ω_s, at each stage of the evolution:

(1) meshed representation: a conforming simplicial mesh $\mathcal{T}$ of D is available in which Ω_f is explicitly discretized as a meshed subdomain; see Figure 3a.
(2) level-set representation: following [12, 104], Ω_f is implicitly described using the level-set method [83]; see Figure 3b. It is the negative subdomain of a function $\phi : D \to \mathbb{R}$ satisfying

$$\forall x \in D, \quad \begin{cases} \phi(x) < 0 & \text{if } x \in \Omega_f, \\ \phi(x) = 0 & \text{if } x \in \Gamma, \\ \phi(x) > 0 & \text{if } x \in \Omega_s. \end{cases} \qquad (3.10)$$

We then consistently alternate between both descriptions of Ω_f depending on their suitability with respect to the ongoing operation. On the one hand, the finite element analyses involved in the resolution of the systems (2.1) to (2.3) and the corresponding adjoint equations (2.11) to (2.13) are carried out by using the meshed representation, which features explicitly the phases Ω_f and Ω_s as meshed subdomains. On the other hand, their motion is tracked using their level-set representation, thus leaving room for topological changes in the course of the evolution.

Let us now outline how this idea is implemented in our shape and topology optimization process. At a given stage of the optimization process, the computational domain D is equipped with a mesh $\mathcal{T}$ in which the current interface Γ (and the corresponding phases Ω_f, Ω_s) is explicitly discretized. A deformation vector field $\boldsymbol{\theta}$ is provided by the null space gradient flow described in section 3.1, such that the new shape $\Gamma_{\boldsymbol{\theta}}$ is "better" (at first order) than the current one Γ. In practice, $\boldsymbol{\theta}$ is given at the vertices of the computational mesh $\mathcal{T}$, cf. Figure 3a. Our mesh evolution algorithm then computes a new mesh $\mathcal{T}'$ of D adapted to the discretization of the deformed configuration $\Gamma_{\boldsymbol{\theta}}$, corresponding to the next iteration, along the following lines:

1. an initial level-set function $\phi_0 : D \to \mathbb{R}$ representing Ω_f is calculated as the signed distance function to Ω_f at the vertices of the mesh $\mathcal{T}$, cf. Figure 3b. Multiple algorithms are available to achieve this, such as fast marching method [99] or the fast sweeping method [92]. In our implementation, we rely on the open-source algorithm Mshdist from one of our previous works [32];
2. the evolution of the shape is accounted for by solving the following advection equation on the mesh $\mathcal{T}$ of D:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t, x) + \boldsymbol{\theta}(t, x) \cdot \nabla \phi(t, x) = 0, & (t, x) \in]0; 1] \times D, \\ \phi(0, x) = \phi_0(x), & x \in D. \end{cases} \qquad (3.11)$$

9

A new level-set function $\phi'(\cdot) := \phi(1, \cdot)$ is obtained, associated with the new fluid and solid domains $\Omega'_f = \Omega_{f,\boldsymbol{\theta}}$ and $\Omega'_s = \Omega_{s,\boldsymbol{\theta}}$, and the new interface $\Gamma' = \Gamma_{\boldsymbol{\theta}} = \partial\Omega'_s \cap \Omega'_f$, see Figure 4a;

3. A new, conforming, and high-quality mesh $\mathcal{T}'$ of D is obtained where Ω_f explicitly appears as a meshed subdomain. This is achieved in two steps, thanks to the open-source platform Mmg [30]:

   (i) the new shape Ω'_f is discretized into $\mathcal{T}$ by splitting its elements according to the zero level set of ϕ'. This part is purely combinatorial and it yields a conforming but poor-quality mesh $\widetilde{\mathcal{T}}$ in which Γ' (and Ω'_f, Ω'_s) explicitly appears as a meshed interface (resp. as meshed subdomains), see Figure 4b;

   (ii) remeshing operations [30, 9, 43, 45] are performed iteratively to improve the quality of the mesh $\widetilde{\mathcal{T}}$ and the approximation of the discrete interface Γ. This yields a high-quality mesh $\mathcal{T}'$, cf. Figure 4c.
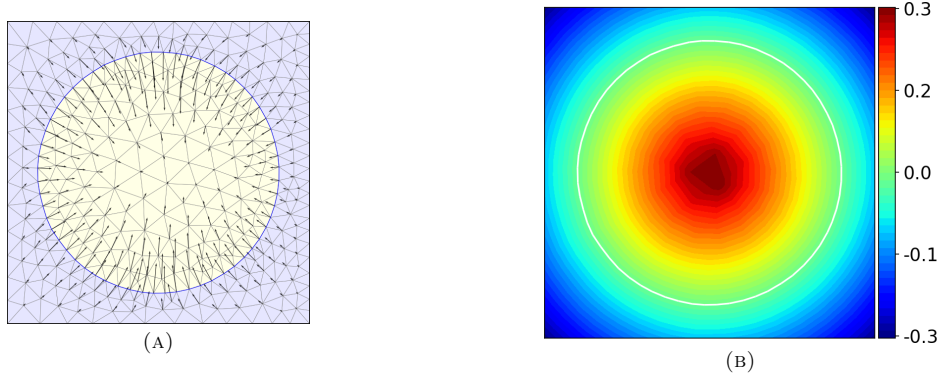

(A)

(B)

FIGURE 3. Initial setting of the level-set based mesh evolution algorithm: (A) a vector field $\boldsymbol{\theta}$ is defined at the vertices of the computational mesh $\mathcal{T}$ of the background domain D in which $\Omega_f \subset D$ is explicitly discretized; (B) a level-set function ϕ_0 representing Ω_f is computed on the mesh $\mathcal{T}$.
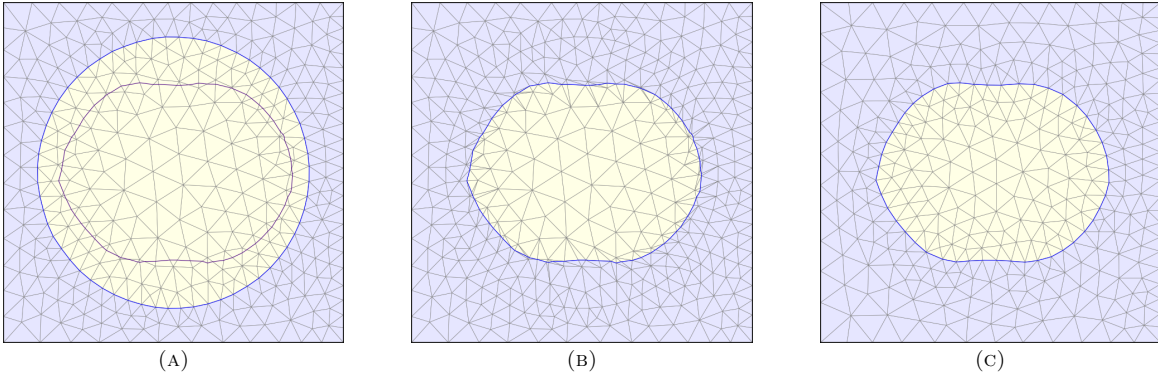

(A)   (B)   (C)

FIGURE 4. Mesh evolution algorithm: (A) the level-set function ϕ_0 associated with the subdomain Ω_f is advected on $\mathcal{T}$ according to $\boldsymbol{\theta}$; (B) a poor-quality mesh $\tilde{\mathcal{T}}$ of the updated domain Ω'_f is obtained by splitting the elements of $\mathcal{T}$ according to the updated level-set function; (C) $\tilde{\mathcal{T}}$ is iteratively remeshed into a new, high-quality mesh $\mathcal{T}'$, which lends itself to finite element analyses.

Remark 2. 1. With this strategy, it is possible to adapt the mesh $\mathcal{T}'$ so that it features small elements near Γ (as a priority near the regions with high curvature), and coarser elements elsewhere.

2. Occasionally, a few low-quality elements may remain in the mesh $\mathcal{T}'$. Nevertheless, in all the treated examples, our finite element resolution strategy proved robust to their presence.

4. Discretization and resolution of the three-dimensional state and adjoint equations

As stressed above, the algorithmic ingredients outlined in section 3 can be used identically for solving both 2D (as illustrated in [43]) and 3D topology optimization problems, in theory. However, the implementation in the three-dimension case requires substantial algorithmic efforts. At first indeed, important differences must be taken into account when it comes to the meshing and mesh adaptation of two- and three-dimensional domains. We do not discuss further these delicate matters, since they are extensively detailed in our previous work [30]. Secondly, one cornerstone of three-dimensional simulations lies in the assembly and in the resolution of large sparse linear systems. This is the main focus of this section.

4.1. Foreword: solving large linear systems arising from the finite element discretization

The computation of the solutions to the physical state equations (2.1) to (2.3) requires to solve various linear or linearized nonlinear systems of the form:

$$Ax = b. \tag{4.1}$$

The matrix A results from the finite element discretization of a linear differential operator, or its linearization when it comes to the Navier–Stokes equations (2.1). Such systems also occur during the resolution of the adjoint problems (2.11) to (2.13) involved in the calculation of shape derivatives, see proposition 1. In a three-dimensional situation, the size of A quickly becomes prohibitively large for the use of exact LU or Cholesky factorizations to solve (4.1), since their algorithmic cost grows superlinearly with respect to the size of A [47, 84]. Classically, at large scale, one resorts instead to iterative methods, such as the conjugate gradient algorithm (CG) or GMRES, whose convergence rate depends crucially on the condition number of A, see e.g. [94]. These methods are very competitive provided that the system (4.1) can be appropriately preconditioned, so that a reasonable number of iterations is required. Unlike exact factorization techniques, which are applied as black box solvers on the coefficient matrix A, the preconditioning stage of iterative solvers must be adapted carefully depending on the physics at hand, as we discuss in the next paragraph.

Before entering into specifics, let us briefly detail our numerical environment. In our applications, we rely on FreeFEM [49] for the finite element discretization of our variational problems, and on PETSc [18, 19] as the linear algebra behind the numerical resolution of the associated linear systems. As we have seen in section 3.2, the computational meshes used in our strategy are adapted to the physical domains: they are calculated owing to the Mmg library (currently, in a sequential manner), and they are then partitioned using standard libraries such as SCOTCH [26] or Metis [62]. After that initial step, assembling finite element matrices, solving linear systems, and evaluating volume or surface integrals is performed in a distributed fashion, see [60] for a more in-depth introduction. Let us mention that several works have recently been conducted in the shape and topology optimization literature, regarding the use of such high performance computing techniques; however, these either rely on non-adaptive structured meshes [1] or refined structured meshes [96]. The state-of-the-art in shape and topology optimization using unstructured, adapted meshes seems rather limited, see [58] for a two-dimensional sequential monophysics example.

4.2. Physics-dependent preconditioners for the resolution of the state and adjoint equations

As we have explained in section 4.1, the use of exact factorizations is ruled out because of the typically large size of the systems involved in three-dimensional settings. Physics-dependent preconditioners must then be tailored to each system (2.1) to (2.3) to ensure that the iterative linear algebra methods involved in their resolution are robust throughout the optimization process and converge rapidly.

4.2.1. Linear elasticity

The discrete counterpart of the linear elasticity equation (2.3) is a symmetric definite positive system of the form (4.1). The smoothed-aggregation Geometric Algebraic Multigrid (GAMG) preconditioner from PETSc is known to be very efficient for solving such systems [3]. To make it robust with respect to variable mesh size and material coefficients, it is properly configured using a matrix blocksize of 3 (i.e., all unknowns are coupled in small dense 3-by-3 blocks) and a near null space of dimension 6 made of the rigid body modes, e.g., 3 translations and 3 rotations. Furthermore, a coarsening threshold of 0.01 is used.

4.2.2. *Heat conduction–convection*

The thermal problem (2.2) arises in a convection-dominated regime. Thus, a simple preconditioner such as the additive Schwarz method (ASM) [105] with one level of algebraic overlap and using exact subdomain solvers proves to be efficient.

4.2.3. *Navier–Stokes equations*

Solving the 3D steady-state incompressible Navier–Stokes equations is more challenging than the previous elasticity and heat conduction problems, partly because the identification of appropriate preconditioners is more delicate.

Our implementation relies on the modified augmented Lagrangian preconditionner (mAL) [21], whose performance at large-scale was recently assessed by using FreeFEM and PETSc [79]. Furthermore, since its implementation is freely available and thanks to the flexibility of FreeFEM and the composability of PETSc solvers, the integration of the mAL preconditioner in our optimization framework is rather straightforward. The cornerstone of this method is the addition of a penalization term for the null-divergence constraint in the variational formulation associated with the nonlinear Navier–Stokes problem (2.1):

find $(\boldsymbol{v}, p) \in V_{\boldsymbol{v},p}(\Gamma)$ such that $\forall (\boldsymbol{w}, q) \in V_{\boldsymbol{v},p}(\Gamma)$,

$$\int_{\Omega_f} \left[\sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v} \cdot \boldsymbol{v} - q \mathrm{div}(\boldsymbol{v}) - p \mathrm{div}(\boldsymbol{w}) + \gamma \mathrm{div}(\boldsymbol{v}) \mathrm{div}(\boldsymbol{w}) \right] \mathrm{d}x = \int_{\Omega_f} \boldsymbol{f}_f \cdot \boldsymbol{w} \mathrm{d}x, \quad (4.2)$$

where we recall that the space $V_{\boldsymbol{v},p}(\Gamma)$ is defined in (2.16), and the coefficient $\gamma > 0$ penalizes the constraint $\mathrm{div}(\boldsymbol{v}) = 0$. The nonlinear problem (4.2) is solved thanks to the Newton method: starting from an initial guess $(\boldsymbol{v}_0, p_0)$, at each step $k \geqslant 0$ of the process, the current couple $(\boldsymbol{v}_k, p_k)$ of velocity and pressure is updated as:

$$\boldsymbol{v}_{k+1} := \boldsymbol{v}_k + \delta \boldsymbol{v}_k, \quad p_{k+1} := p_k + \delta p_k,$$

where the increment $(\delta \boldsymbol{v}_k, \delta p_k)$ is computed by solving the linearized version of (4.2) around $(\boldsymbol{v}_k, p_k)$, the so-called Oseen problem:

find $(\delta \boldsymbol{v}_k, \delta p_k) \in V_{\boldsymbol{v},p}(\Gamma)$ such that $\forall (\boldsymbol{w}, q) \in V_{\boldsymbol{v},p}(\Gamma)$,

$$\int_{\Omega_f} \left[\sigma_f(\delta \boldsymbol{v}_k, \delta p_k) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}_k \cdot \delta \boldsymbol{v}_k + \rho \boldsymbol{w} \cdot \nabla (\delta \boldsymbol{v}_k) \cdot \boldsymbol{v}_k - q \mathrm{div}(\delta \boldsymbol{v}_k) - \delta p_k \mathrm{div}(\boldsymbol{w}) \right] \mathrm{d}x$$

$$+ \int_{\Omega_f} \gamma \mathrm{div}(\delta \boldsymbol{v}_k) \mathrm{div}(\boldsymbol{w}) \mathrm{d}x = \int_{\Omega_f} \boldsymbol{f}_f \cdot \boldsymbol{w} \mathrm{d}x. \quad (4.3)$$

In our implementation, the initial guess $(\boldsymbol{v}_0, p_0)$ is the solution to the Stokes counterpart problem to (4.2), obtained by setting $\rho = 0$. The difficult part of the method is the resolution of the discretization of (4.3), which is of the form:

$$A_{\mathrm{Oseen}} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \quad (4.4)$$

where A is the matrix discretizing the bilinear form of (4.3):

$$(\delta \boldsymbol{v}_k, \boldsymbol{w}) \mapsto \int_{\Omega_f} \left[\sigma_f(\delta \boldsymbol{v}_k, \delta p_k) : \nabla \boldsymbol{w} + \rho \boldsymbol{w} \cdot \nabla \boldsymbol{v}_k \cdot \delta \boldsymbol{v}_k + \rho \boldsymbol{w} \cdot \nabla (\delta \boldsymbol{v}_k) \cdot \boldsymbol{v}_k + \gamma \mathrm{div}(\delta \boldsymbol{v}_k) \mathrm{div}(\boldsymbol{w}) \right] \mathrm{d}x,$$

and the block B is the discretization of the divergence operator $(\boldsymbol{w}, q) \mapsto \int_{\Omega_f} q \mathrm{div} \boldsymbol{w} \mathrm{d}x$. Following [79], a lower-triangular preconditioner M_{Oseen} for A_{Oseen} is built by using the PETSc FieldSplit feature, which roughly speaking amounts to split the velocity and pressure fields:

$$M_{\mathrm{Oseen}} := \begin{bmatrix} A & 0 \\ B & S \end{bmatrix}. \quad (4.5)$$

The action of the inverse of the pressure Schur complement $S = -BA^{-1}B^T$ is computed from the user-supplied approximation $S^{-1} = -(\gamma + 1/\mathrm{Re})M_p$, where M_p is the mass matrix, corresponding to the bilinear

form $(p, q) \mapsto \int_{\Omega_f} pq\mathrm{d}x$ and $\mathrm{Re} = \rho/\nu$ is the Reynolds number. The action of the inverse of A^{-1} on a vector is computed by further splitting A into 3 diagonal blocks (one for each component of the velocity), and solving each subsystem by means of the ASM with exact subdomain solvers for the right-preconditioned GMRES [95]. As the velocity preconditioners are variable, the outer iterative method is the flexible GMRES [93].

4.2.4. *Adjoint linear systems*

On the one hand, little to no modification of the preconditioner settings for the elasticity and thermal problems is required for solving efficiently the respective adjoint systems (2.11) and (2.12). Thus, GAMG (resp. ASM) is still used to this purpose with the same parameters as in section 4.2.1 (resp. section 4.2.2).

The treatment of the adjoint fluid system (2.13) is slightly different to that of the direct Navier–Stokes problem described in section 4.2.3. Indeed, although although the linear system associated to (2.13) involves the transpose of the Oseen matrix (4.4), (2.13) implies that the divergence $\mathrm{div}(\boldsymbol{w})$ of the fluid adjoint variable $\boldsymbol{w}$ may be nonzero if the objective function $J(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma), T(\Gamma), \boldsymbol{u}(\Gamma))$, and thus the transported function $\mathfrak{J}$ of (2.9) depends on the pressure variable p. For this reason, the mAL preconditioner cannot be used anymore, as its stability crucially relies on the penalization term $\gamma\mathrm{div}(\boldsymbol{v})\mathrm{div}(\boldsymbol{w})$ in (4.2), which is supposed to be close to zero.

For this reason, another PETSc FieldSplit configuration is used. The velocity and pressure fields are still split, but instead of using a multiplicative block Gauss–Seidel FieldSplit type, a Schur complement preconditioner is built by PETSc. This time, the velocity field is not split according to its components in the canonical basis of $\mathbb{R}^3$, but the ASM with exact subdomain solvers is still used to right-precondition the GMRES procedure involved in the approximation of the inverse of the velocity block A. For the pressure block, at most 5 iterations of the Jacobi-preconditioned CG are computed.

4.2.5. *Comparative performance of various stages of the parallel finite element implementation*

We evaluate the computational efficiency of our numerical framework in the context of the lift-drag design test case of section 5.3 where only the fluid physics is involved.

Using three different executions, corresponding to different numbers of processes, we report in Table 1 the running times corresponding to the main operations performed during the first iteration of our optimization algorithm. The number of vertices of the considered computational mesh is $33, 500$ in the total domain and $30,180$ in the fluid domain. This configuration corresponds to a rather 'small' problem but it is considered to allow for the comparison with a run on a single process. More precisely, we examine the following operations of our overall implementation which are accelerated by parallel computing:

(1) all steps required for solving the state equations (2.1) to (2.3);
(2) the computation of the values of the objective and constraint functions;
(3) all steps required for computing the shape derivatives of these functionals, including the resolution of the adjoint systems;
(4) the resolution of identification problems of the form (3.9) which feature the inversion of a linear elliptic problem.

We note that the most computationally expensive tasks consists in the solution of the Navier-Stokes system (2.1) and of the adjoint problem (2.13), as well as the various finite element matrix or vector assembly steps. As already noticed in previous works, e.g. [22], we observed in several of our test cases that iterative solvers require often more iterations to converge for the adjoint problem than for the primal. All things considered, the scaling obtained is quite satisfactory and it allows us to run multiphysics shape optimization test cases in reasonable CPU time.

5. Three-dimensional shape and topology optimization test cases

In this section, we treat five instances of the program (2.5). The first three sets of examples involve only a single physics, which is helpful to validate our implementation before passing to more challenging coupled physics applications.

In section 5.1, we examine the optimization of the shape of mechanical structures subject to either bending or torsional loads in the regime of linear elasticity.

Action	1 process	16 processes
Partitioning	0.91	3.2
Build Stokes matrix and RHS	13.55	1.28
Solve Stokes problem	22.75	2.85
Solve Oseen linear problems for three Newton iterations	383	13.2
Assemble FE matrices for three Newton iterations	153	15.6
Computing objective function Lift	13.9	1.48
Computing drag constraint Drag	9.3	0.97
Assembling the adjoint fluid matrix	74.23	6.66
Assembling Riesz identification matrix	1.04	0.09
Assembling adjoint RHS for the objective function Lift	12.58	1.24
Solved adjoint fluid problem (5.6) for the Lift constraint	36.09	14.09
Assembled shape derivative of the Lift functional	0.4	0.08
Identification of the shape derivative to a gradient	0.64	0.49
Total	820.53	96.53

TABLE 1. Runtimes all finite element operations performed at the first iteration of the aerodynamic design test case of section 5.3 performed on Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz.

The next section 5.2 considers an optimal design problem in pure heat conduction. The physics at play is admittedly the least complicated to solve among the four considered situations since it involves only a scalar elliptic problem; nevertheless, the obtained optimized design is very intricate and it clearly illustrates the efficiency of our mesh evolution method and the robustness of the remeshing library Mmg.

The third context of interest is that of shape and topology optimization in fluid mechanics. In section 5.3, we tackle the very classical problem of finding optimal aerodynamic designs with respect to the lift and drag forces induced by a fluid. This problem has been the object of much effort in the literature, as in e.g. [55, 89]. However, these contributions most often consider industrial contexts featuring very high Reynolds numbers and where the shape to optimize is parameterized by a small number of parameters. This makes sense because very small design update can lead to a substantial increase of performance. Often, automatic differentiation is used rather than Hadamard's shape derivatives in order to obtain the sensitivity to these parameters in the context of the solution of the physics with industrial codes. By contrast, we demonstrate that the application of our shape and topology optimization method, relying on analytic Hadamard's shape derivatives, allows to compute optimized aerodynamic designs, for moderate Reynolds numbers, without resorting to any parameterization of the shape.

The last two test cases involve true coupled physics applications involving two physics at a time. Section 5.4 is concerned with the shape and topology optimization of a three-dimensional fluid–structure interacting system: a vertical plate is pushed down by a fluid. The problem is then to find a distribution of solid material around the plate in order to make the whole structure as little compliant as possible.

Finally, section 5.5 shows an application of our method to a three-dimensional shape and topology optimization problem in convective heat transfer. These last two test cases involve meshes containing up to 300,000 vertices in both the fluid and solid domains, which results in linear systems featuring more than 5×10^6 degrees of freedom.

All these results can still be considered as preliminary, since many improvements could be considered in order to reach much larger, industrial size problems: one of the limiting factor at the present moment lies in the fact that the remeshing step of the optimization algorithm is still sequential and very expensive, see

the running CPU times reported below in Table 2. However, to the best of our knowledge, there is currently no freely available library for performing unstructured mesh adaptation according to a level-set function in a distributed fashion.

5.1. **Cantilever beam subject to bending or torsional loads**

We start by reproducing the classical benchmark test case of a three-dimensional cantilever beam subject to either a bending or a torsional load. The computational domain D is a box of dimensions $2 \times 1 \times 1$. The optimized solid structure $\Omega_s \subset D$ is fixed at four squares of size 0.3×0.3 located on the left-hand side of the boundary ∂D as depicted on Figure 5. A force $\boldsymbol{g}$ is applied on a disk-shaped region at the center of the right-hand side of ∂D. We consider two situations:

- bending load: $\boldsymbol{g}$ is vertical and pointing downward: $\boldsymbol{g} := -\boldsymbol{e}_y$ where $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ denotes the canonical basis of $\mathbb{R}^3$;
- torsional load: $\boldsymbol{g}$ corresponds to a torsion force field:

$$\boldsymbol{g} := \begin{bmatrix} 0 & \dfrac{(z - 0.5)}{\sqrt{(y - 0.5)^2 + (z - 0.5)^2}} & -\dfrac{(y - 0.5)}{\sqrt{(y - 0.5)^2 + (z - 0.5)^2}} \end{bmatrix}^T .$$

In both cases, we aim to minimize the compliance of the structure Ω_s under a volume constraint:

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

$$\text{s.t.} \quad \mathrm{Vol}(\Omega_s) := \int_{\Omega_s} \mathrm{d}x = V_T, \tag{5.1}$$

where $\boldsymbol{u}(\Omega_s)$ (or simply $\boldsymbol{u}$ for brevity) is the solution to the linear elasticity system (2.3) in the above context, and V_T is a target volume, set to respectively 0.15 and 0.08 for the bending and torsion test cases.
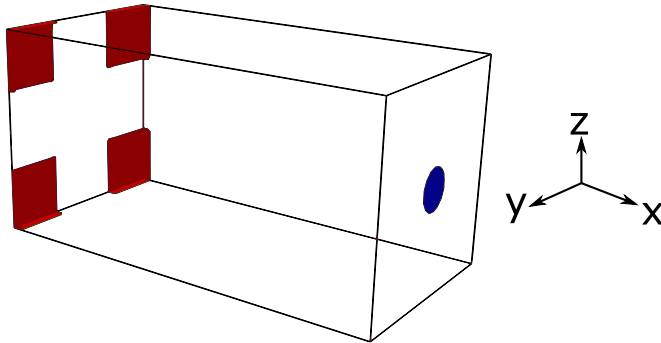
FIGURE 5. Setting of the three-dimensional cantilever test case of section 5.1 subjected to a bending load. The red regions correspond to four square-shaped Dirichlet boundaries to which the whole structure is fixed, and the blue disk is the region of ∂D supporting Neumann boundary conditions (where either a bending or a torsion force field is applied).

5.1.1. *Bending test case*

We first deal with the bending test case, which has already been treated with the level-set mesh evolution algorithm in our previous work [9], with a much smaller resolution however: the initial mesh in the present study contains 108,605 vertices, whereas it only contained 18,081 vertices in the aforementioned article. The present higher resolution allows the capture of very thin plate-like structures which are rarely seen when using too coarse meshes.

The numerical results are shown on Figure 6, and the convergence histories for the objective and constraint functions are provided in Figure 7. These plots illustrate the use of remeshing in our topology optimization methodology. Remarkably, a high-quality mesh is available at each stage of the optimization process, while the shape undergoes multiple topological changes in the course of the evolution.
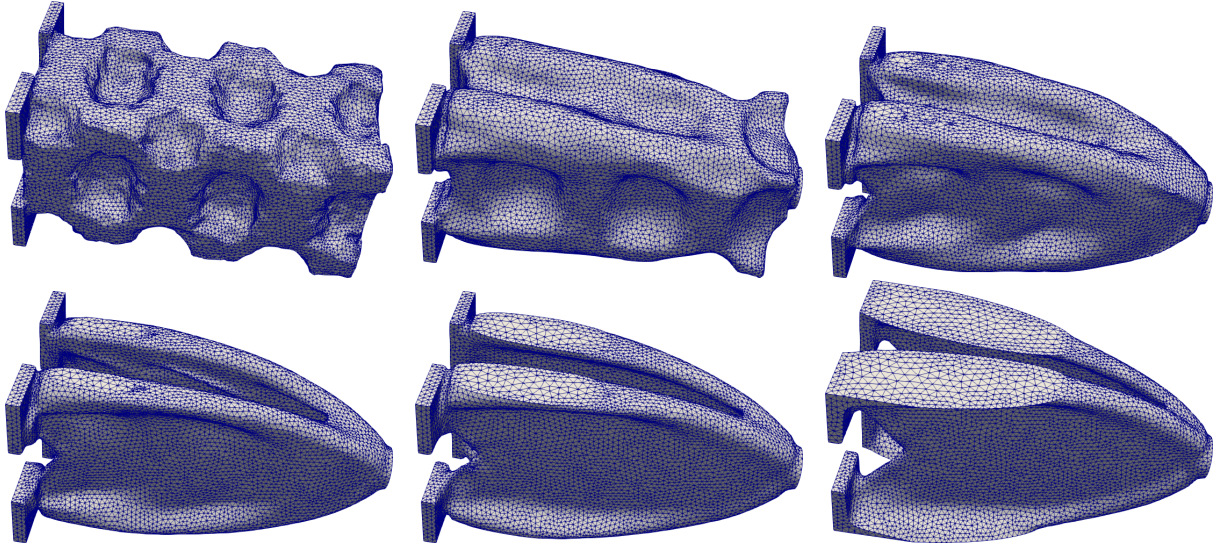
15

FIGURE 6. From top left to bottom right: iterations 0, 5, 10, 20, 40 and 1250 of the optimization of a three-dimensional cantilever subjected to a bending load, cf. section 5.1.
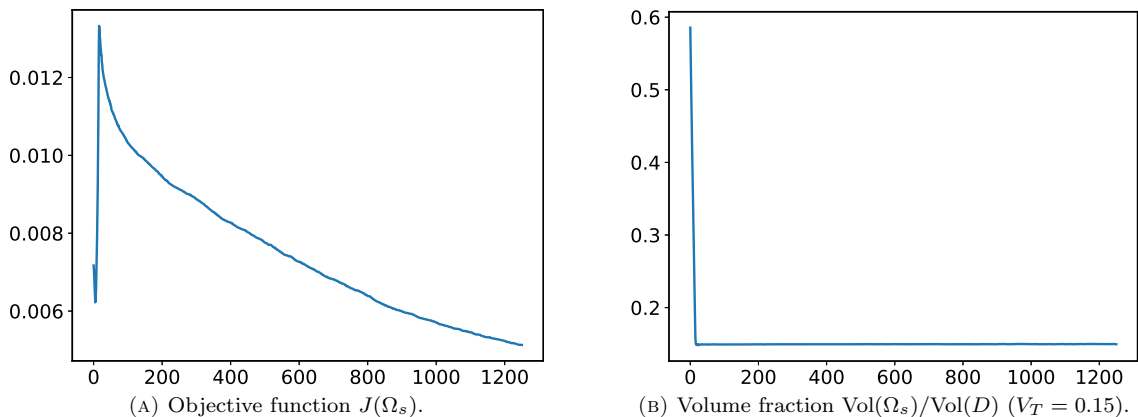

(A) Objective function $J(\Omega_s)$. (B) Volume fraction $\text{Vol}(\Omega_s)/\text{Vol}(D)$ $(V_T = 0.15)$.

FIGURE 7. Convergence histories for the 3D cantilever test cases subjected to a bending load of section 5.1.1.

5.1.2. *Torsion test case*

Optimization results for the torsion test case are presented on Figure 8; the convergence histories for the objective and constraint functions are plotted on Figure 9. It is interesting to note that the optimized shape is utterly different from that obtained in the bending test case in section 5.1.1, the outer part of the structure showing multiple holes in the present situation.

5.1.3. *On the influence of the remeshing method*

Remarkably, the topology optimized designs obtained with our remeshing algorithm feature plate-like structures not thicker than one or two mesh elements (as illustrated on Figure 2). These results are quite original when compared to those obtained with more classical SIMP or level-set methods, where the lower resolution imposed by the uniform size of the grid has the tendency to generate structures featuring rather holes and bars [11, 98, 1].

Note that, in the two numerical examples of Figs. 7 and 9, convergence is not fully attained at the last iteration (respectively 1,250 and 3,500 for the bending and torsion test cases), since the objective function is still decreasing slightly. As illustrated in Figs. 6 and 8, the most essential features of the final shape are
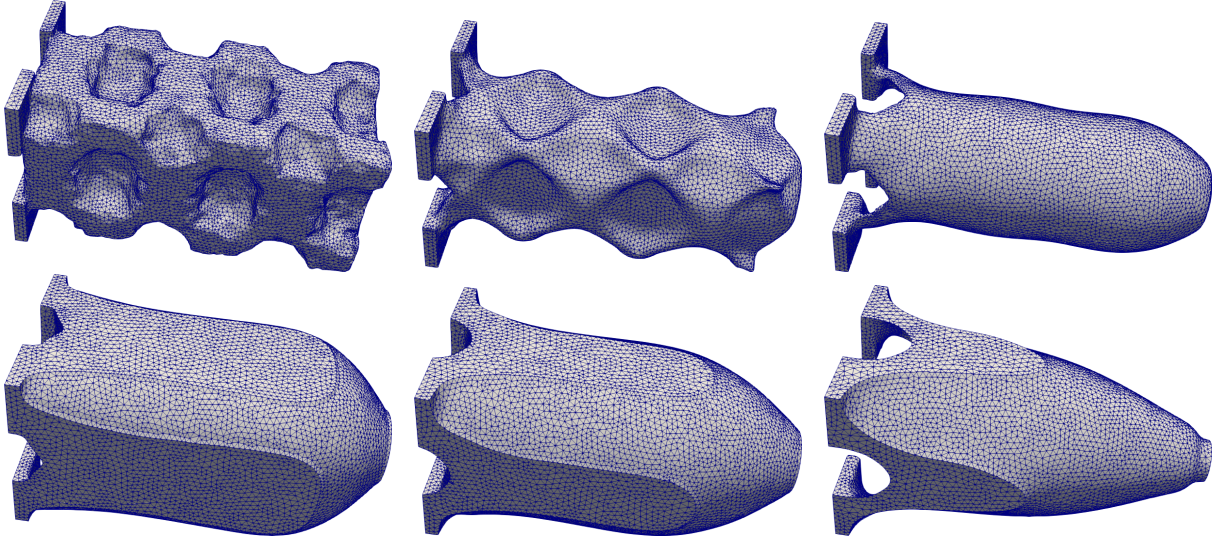
FIGURE 8. From top left to bottom right: iterations 0, 10, 30, 1,000, 2,000, and 3,500 of the optimization of a three-dimensional cantilever subjected to a torsional load, cf. section 5.1.2.

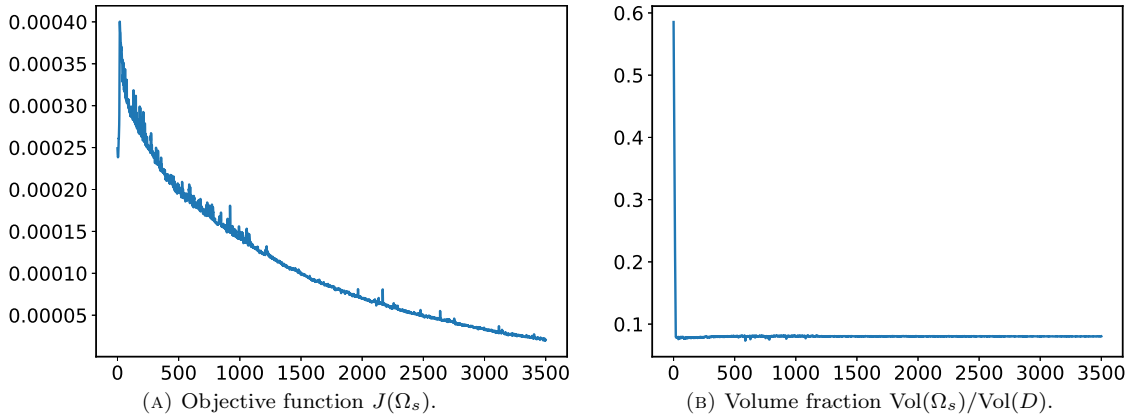(A) Objective function $J(\Omega_s)$.

(B) Volume fraction $\mathrm{Vol}(\Omega_s)/\mathrm{Vol}(D)$.

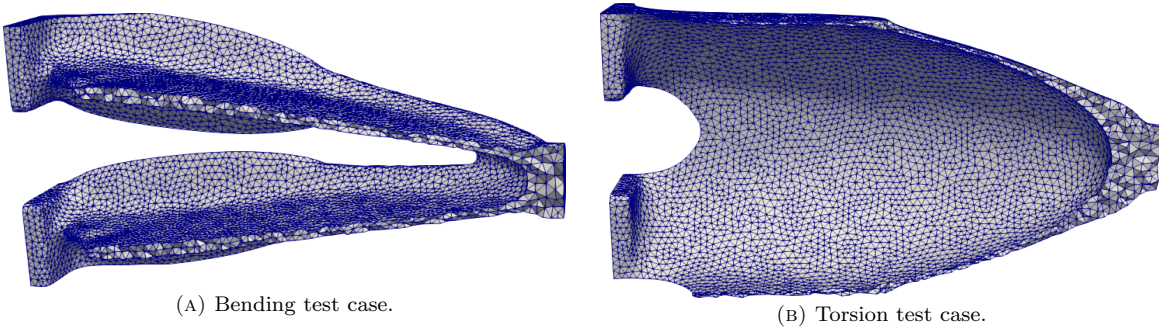FIGURE 9. Convergence histories for the 3D cantilever test case subjected to a torsional load of section 5.1.2.

(A) Bending test case.

(B) Torsion test case.

FIGURE 10. Cut of the optimized designs obtained in sections 5.1.1 and 5.1.2. Our remeshing algorithm is able to capture very thin plate-like structures not thicker than one or two mesh elements.

captured quite quickly (in about 40 iterations), however we found that the optimizer still manages to improve the shape very slowly after many iterations. Our experience suggests this behavior could be explained by the facts that (i) the optimization landscape is locally rough (convergence is not improved by taking larger steps) and (ii) when there are no more topology changes, our remeshing strategy works like a mesh adaptation and allows for very small scale changes which can have a definite influence on the objective function. Maybe it is a manifestation of the ill-posed character of shape optimization, whereby smaller and smaller details can improve the objective function. In any case, it is an important issue which deserves further investigations and cannot be done here by lack of space (we plan to deal with it in a future work).

5.2. **Optimal design for pure thermal heat conduction**

We now focus on the optimization of a pure heat conduction system: only the thermal equation (2.2) is solved. The setting is represented on Figure 11. The hold-all domain D is a box with size $1 \times 1 \times 1$; it arises as the disjoint reunion of a phase Ω_s filled by a material with low conductivity $k_s = 1$ and its complement Ω_f with a material with high conductivity $k_f = 100$. A Dirichlet boundary condition is imposed on a small square of size 0.4×0.4 at the bottom face of ∂D where the temperature $T = 0$ is imposed. All other external boundaries of the cube D are adiabatic, i.e., $\partial T / \partial \boldsymbol{n} = 0$. The whole domain is heated with a source
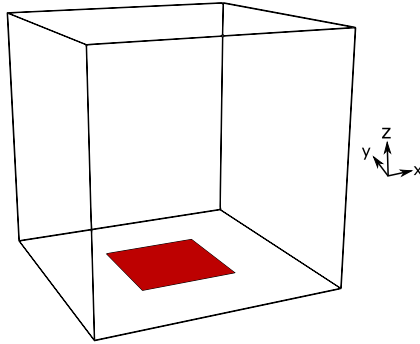

FIGURE 11. Setting for the pure conduction test case of section 5.2. The cube is heated uniformly and the temperature is prescribed to $T = 0$ at the red square of the bottom face. The other boundaries are adiabatic.

$Q_s = Q_f = 10^4$ and the goal is to find the shape of the interface $\Gamma = \partial \Omega_s \cap \partial \Omega_f$ between the two materials which minimizes the average temperature over D under a constraint on the volume of the highly conducting phase Ω_f:

$$\min_{\Gamma} \quad J(\Gamma, T(\Gamma)) := \int_D T \mathrm{d}x$$
$$\text{s.t.} \quad \text{Vol}(\Omega_f) = V_T,$$

(5.2)

where the target volume V_T equals 0.05.

Numerical results are reported on Figs. 12 and 13, and the convergence histories of the optimization process are plotted on Figure 14. The resolution of the mesh of D varies from 63,761 vertices for the initial design to 206,464 vertices for the final design. Remarkably, the remeshing software Mmg and our optimization method are able to capture sheet-like structures not thicker than one or two times the size of an element of the mesh.

5.3. **Lift–drag topology optimization for aerodynamic design**

This section investigates three-dimensional shape and topology optimization for lift–drag problems in the field of aerodynamic design. The lift is the vertical force generated by a flow around an obstacle (e.g. the wing of a plane), keeping it afloat; it is usually defined as a surface integral involving the (fluid) normal stress tensor. The drag is the energy dissipated by the fluid around the obstacle, causing it to slow down. Lift–drag shape optimization, that is, the question of finding designs with large lift and low drag insofar as possible, is a very classical problem which has been the aim of a very large amount of contributions
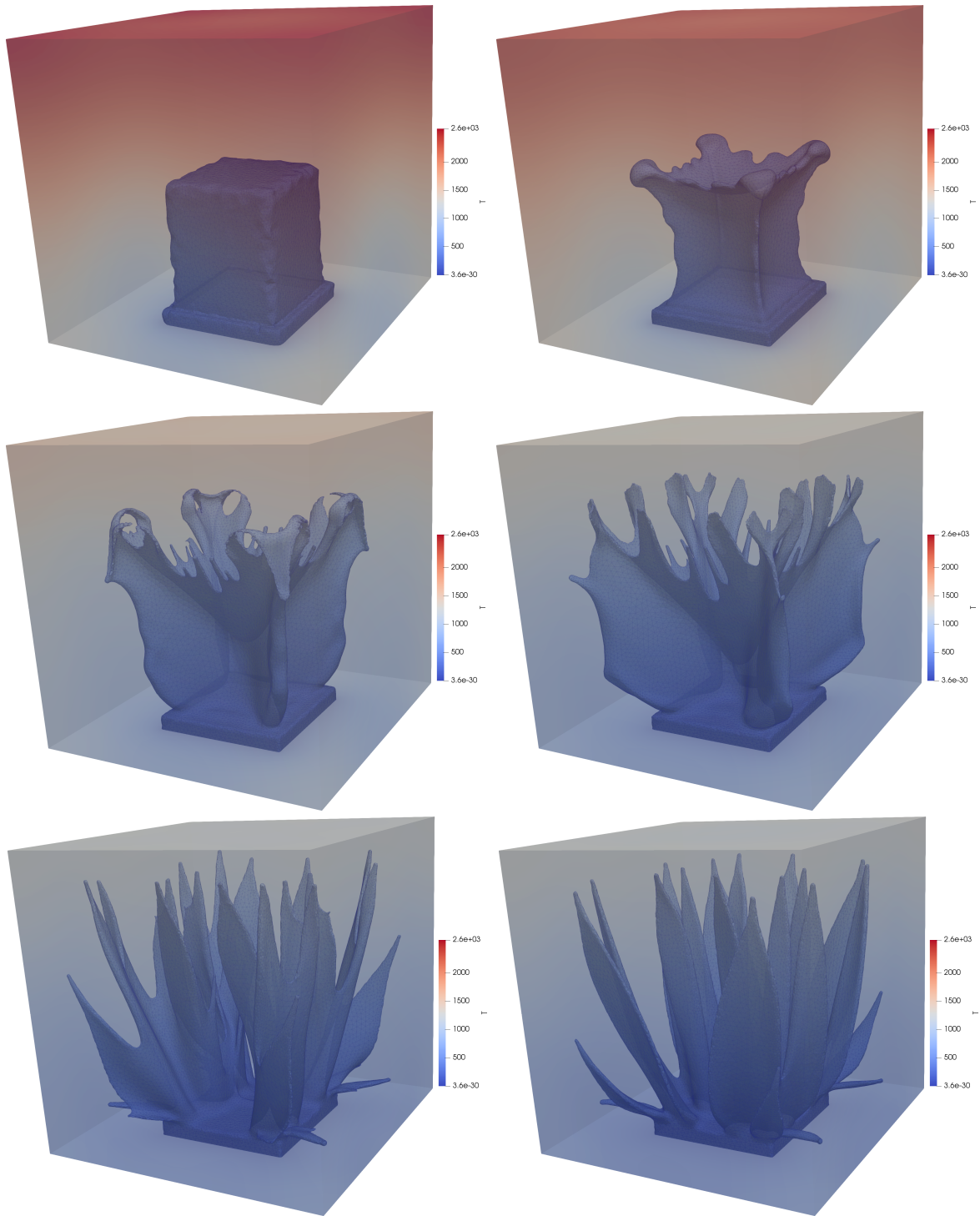
FIGURE 12. From top left to bottom right: iterations 0, 5, 15, 30, 100, and 258 of the optimization of the three-dimensional heat conduction system considered in section 5.2.
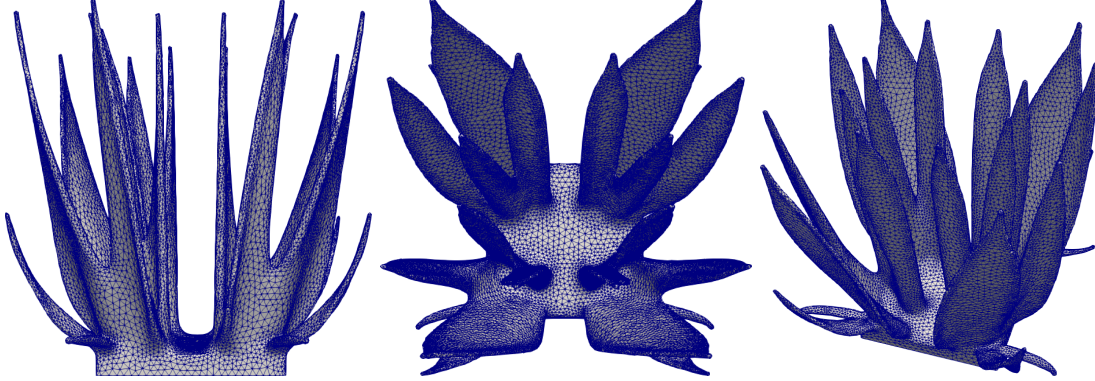
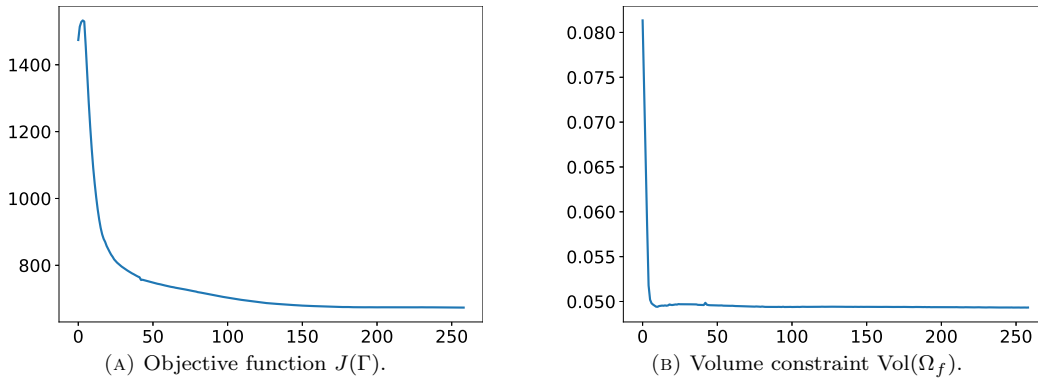FIGURE 13. Different three-dimensional views of the optimized design for the heat conduction test case of section 5.2.

(A) Objective function $J(\Gamma)$.

(B) Volume constraint $\mathrm{Vol}(\Omega_f)$.

FIGURE 14. Convergence histories for the three-dimensional heat conduction test case of section 5.2.

[87, 78, 55, 56, 40, 68, 48]. However, these references have considered situations very close to realistic applications where

(1) the physics is more challenging than in our case, featuring for instance high Reynolds numbers, or a compressible velocity field;
(2) the shape design is usually described by means of a few CAD parameters to optimize;
(3) very small updates of the shape may lead to substantial gains of efficiency.

In this context, it makes sense to seek for improved geometries by means of very small deformations of the proposed CAD design. Few works have actually tried to apply shape and topology optimization techniques to such lift–drag problems, where the design shape is allowed to deform freely: we are essentially aware of [27, 63, 46, 103]. In what follows, we treat a lift–drag optimization problem with the method of Hadamard and our topology optimization framework on a three-dimensional example featuring a moderate Reynolds number Re = 200. We are not aware, to the best of our knowledge, of analogous results in this three-dimensional setting.

We start this study by describing briefly in section 5.3.1 the computation of the shape derivative of the lift functional. Although this question has already been considered in several works of the literature, the calculation and the numerical implementation of the resulting formulas do not seem completely standard to us; we propose a special treatment based on a classical idea of [20, 39]. We then present in section 5.3.2 a three-dimensional example of an optimized design for an instance of the lift–drag problem.

5.3.1. *Shape derivatives of the lift functional*

The computational domain $D = \mathbb{R}^3$ features a liquid phase Ω_f flowing around a solid obstacle $\Omega_s \subset\subset D$, see Figure 15a. The notation convention is that of section 2: the boundary ∂D reads: $\partial D = \partial \Omega_D \cup \partial \Omega_N$, where

the flow is entering the domain from the Dirichlet boundary $\partial\Omega_f^D$ with given velocity $\boldsymbol{v} = \boldsymbol{v}_0$ and exits the domain with a zero normal stress boundary condition $\sigma_f(\boldsymbol{v}, p)\boldsymbol{n} = 0$ on $\partial\Omega_f^N$. The remaining part of the fluid boundary $\partial\Omega_f$ is the solid interface $\Gamma = \partial\Omega_f\backslash(\partial\Omega_f^D \cup \partial\Omega_f^N) = \partial\Omega_s$, which is to be optimized.

The lift generated by the obstacle Ω_s is the total force exerted by the fluid Γ in the vertical y-direction:

$$\text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := -\int_\Gamma \boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{n}\mathrm{d}s, \tag{5.3}$$

where we recall that $\sigma_f(\boldsymbol{v}, p) = 2\nu e(\boldsymbol{v}) - p\mathrm{I}$, and we have used the notation $\boldsymbol{a} \cdot M \cdot \boldsymbol{b} := \boldsymbol{a}^T M \boldsymbol{b}$ for any vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ and matrix $M \in \mathbb{R}^{3\times3}$. Note that the minus sign accounts for our convention of section 2 whereby the normal $\boldsymbol{n}$ is pointing outward the fluid domain. Several authors have considered the optimization of the lift functional, using for instance surrogate models [69], control points [64], the SIMP method and a different objective functional [63], and a time varying setting [90, 82, 71], see also [102] for a numerical analysis of the shape derivative of this functional. In order to calculate this shape derivative, we propose an alternative method, which relies on a reformulation of the lift functional (5.3) as a volume integral. Let $\mathcal{X} \in H^1(\Omega_f)$ be any extension of the constant function 1 from Γ to the whole fluid domain Ω_f. In our particular implementation, $\mathcal{X}$ is obtained as the solution to the following Poisson problem:

$$\begin{cases} -\Delta\mathcal{X} = 0 \text{ in } \Omega_f \\ \quad\mathcal{X} = 1 \text{ on } \Gamma \\ \quad\mathcal{X} = 0 \text{ on } \partial\Omega_f\backslash\Gamma. \end{cases} \tag{5.4}$$

The function $\mathcal{X}\boldsymbol{e}_y$ is an extension of the vector field $\boldsymbol{e}_y$ from Γ to Ω_f, vanishing on $\partial\Omega_f\backslash\Gamma$. this enables to rewrite $\text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$ as a volume integral by using integration by parts:

$$\begin{aligned} \text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) &= -\int_\Gamma \mathcal{X}\boldsymbol{e}_y \cdot \sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{n}\mathrm{d}s \\ &= -\int_{\Omega_f} \mathrm{div}(\mathcal{X}\sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{e}_y)\mathrm{d}x \\ &= -\int_{\Omega_f} (\nabla\mathcal{X} \cdot \sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{e}_y + \mathcal{X}\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) \cdot \boldsymbol{e}_y)\mathrm{d}x \\ &= \int_{\Omega_f} (\mathcal{X}\boldsymbol{f}_f \cdot \boldsymbol{e}_y - \rho\mathcal{X}\boldsymbol{e}_y \cdot \nabla\boldsymbol{v} \cdot \boldsymbol{v} - \nabla\mathcal{X} \cdot \sigma_f(\boldsymbol{v}, p) \cdot \boldsymbol{e}_y)\mathrm{d}x, \end{aligned} \tag{5.5}$$

where we have used the state equation $-\mathrm{div}(\sigma_f(\boldsymbol{v}, p)) + \rho\nabla\boldsymbol{v}\boldsymbol{v} = \boldsymbol{f}_f$ to obtain the last line. This reformulation is rather classical and is often used in numerical applications, since it is known to yield a more accurate evaluation of the lift functional [20, 39]. Equation (5.5) can now be differentiated with respect to the shape following a methodology very similar to that presented in [43]. The detailed computation is available in [42], Proposition 6.1. Here, we limit ourselves with the statement of the result:

Proposition 2. *Let the adjoint fluid variable* $(\boldsymbol{w}, q) \in V_{\boldsymbol{v},p}(\Gamma)$ *be the solution to*

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

$$\int_{\Omega_f} \left(\sigma_f(\boldsymbol{w}, q) : \nabla\boldsymbol{w}' + \rho\boldsymbol{w} \cdot \nabla\boldsymbol{w}' \cdot \boldsymbol{v} + \rho\boldsymbol{w} \cdot \nabla\boldsymbol{v} \cdot \boldsymbol{w}' - q'\mathrm{div}(\boldsymbol{w})\right)\mathrm{d}x = \frac{\partial\text{Lift}}{\partial(\boldsymbol{v}, p)}(\boldsymbol{w}', q'), \tag{5.6}$$

where $V_{\boldsymbol{v},p}(\Gamma)$ *is the functional space defined in* (2.16). *The lift functional* $\Gamma \mapsto \text{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma))$ *is differentiable with respect to the domain and its shape derivative reads:*

$$\left.\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=0} \left[\text{Lift}(\Gamma_{\boldsymbol{\theta}}, \boldsymbol{v}(\Gamma_{\boldsymbol{\theta}}), p(\Gamma_{\boldsymbol{\theta}}), T(\Gamma_{\boldsymbol{\theta}}), \boldsymbol{u}(\Gamma_{\boldsymbol{\theta}}))\right](\boldsymbol{\theta})$$

$$= \int_\Gamma \left[\boldsymbol{f}_f \cdot \boldsymbol{e}_y + 2\nu e(\boldsymbol{v}) : (\nabla\mathcal{X} \otimes \boldsymbol{e}_y) + \boldsymbol{f}_f \cdot \boldsymbol{w} + \sigma_f(\boldsymbol{w}, q) : \nabla\boldsymbol{v}\right]\boldsymbol{\theta} \cdot \boldsymbol{n}\mathrm{d}s. \tag{5.7}$$

The above formula, in surface form, is easily implemented in our shape and topology optimization framework summarized in section 2. It is at the basis of the solution of the test case presented in the next paragraph.

Remark 3. Similarly as in the point (2) of remark 1, there exists also a volume expression of the shape derivative (5.7) which is slightly lengthier (see Proposition 6.1 in [42] for the full details), and which yields the same numerical optimum in this case.

5.3.2. *Numerical test case*

We consider the problem of maximizing the lift generated by a flow obstacle $\Omega_s \subset\subset D$ under an inequality constraint on the drag so that the obstacle remains aerodynamic. In addition, the volume occupied by the obstacle and the location of its center of mass are prescribed:

$$
\begin{aligned}
\min \quad & -\operatorname{Lift}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \\
\text{s.t.} \quad & \begin{cases} \operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leqslant \operatorname{Drag}_0 \\ \operatorname{Vol}(\Omega_f) = V_T \\ \boldsymbol{X}(\Omega_s) := \dfrac{1}{|\Omega_s|} \displaystyle\int_{\Omega_s} \boldsymbol{x}\mathrm{d}x = \boldsymbol{x}_0. \end{cases}
\end{aligned}
\tag{5.8}
$$

The drag functional is defined by

$$
\operatorname{Drag}(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) := \int_{\Omega_f} \sigma_f(\boldsymbol{v}, p) : \nabla \boldsymbol{v}\mathrm{d}x = \int_{\Omega_f} 2\nu e(\boldsymbol{v}) : e(\boldsymbol{v})\mathrm{d}x.
$$

The computation of its shape derivative is classical, see e.g. [42, 33, 52].

We solve this problem in three space dimensions: the hold-all domain is the box $D = [0,1]^3$. A flow is entering with a velocity $\boldsymbol{v}_0 = \boldsymbol{e}_x$ from the left-hand side of D. A free-slip boundary condition $\boldsymbol{v} \cdot \boldsymbol{n} = 0$ is imposed on the other boundaries of the cube. The physical parameters are still set to $\nu = 1/200$ and $\rho = 1$ corresponding to $\operatorname{Re} = \frac{\rho H v_{\max}}{\nu} = 200$, the characteristic length of D being $H = 1$. The volume target is set to $V_T = 0.01$.

The imposed upper bound Drag_0 is set to $1.5 \cdot \operatorname{Drag}^* = 0.0456$, where $\operatorname{Drag}^* = 0.0304$ is the optimized value obtained in the resolution of the minimum drag problem subject to the same volume and position constraints (computation not reported in this paper).

The optimized shape with associated velocity profile is shown on Figure 15. The final fluid mesh features 54,299 vertices. The finite element problems associated with the Navier–Stokes and adjoint equations are solved by using the parallel computing technique described in section 4 with 12 processes. Convergence histories are shown on Figure 16. A single solution of the state equations, including discretization and application of the nonlinear Newton solver, takes approximately 2 minutes. The computation of the shape derivative, including adjoing system solutions, takes approximately the same time. Each remeshing step performed sequentially is achieved approximately in one minute.

5.4. A three-dimensional fluid–structure interaction test case

Our first multiphysics test case is concerned with a moderately large-scale fluid–structure interaction problem. The situation is that on Figure 17: the computational domain is the box $D = [0,4] \times [0,1] \times [0,1]$. It is the disjoint reunion of a fluid phase Ω_f and a mechanical structure Ω_s. A fluid is entering D from its left boundary with a velocity $\boldsymbol{v}_0 = y\boldsymbol{e}_x$. A no-slip boundary condition $\boldsymbol{v} = 0$ is prescribed on the bottom face of the domain, while the top and side faces bear a free-slip boundary condition $\boldsymbol{v} \cdot \boldsymbol{n} = 0$. The flow exits the domain with a zero normal stress boundary condition $\sigma_f(\boldsymbol{v}, p)\boldsymbol{n} = 0$. The mechanical structure $\Omega_s \subset D$ is fixed on a square patch of the bottom face (i.e. the Dirichlet boundary condition $\boldsymbol{u} = 0$ is imposed in this region) and it is subjected to the stress induced by the fluid; namely, $\boldsymbol{u}$ is the solution of (2.3)). A vertical plate is set as a non-optimizable part of the mechanical structure, as well as a small layer above the bottom Dirichlet boundary. In this context, our goal is to find how to distribute additional material in order to make the structure Ω_s as rigid as possible. The problem features of course a volume constraint on the mechanical structure, so that it reads

$$
\begin{aligned}
\min_{\Gamma} \quad & J(\Gamma, \boldsymbol{u}(\Gamma)) := \int_{\Omega_s} Ae(\boldsymbol{u}) : e(\boldsymbol{u})\mathrm{d}x \\
\text{s.t.} \quad & \operatorname{Vol}(\Omega_s) = V_T.
\end{aligned}
\tag{5.9}
$$

(A) Initial shape.

(B) Optimized design.

(C) Other views of the optimized design.
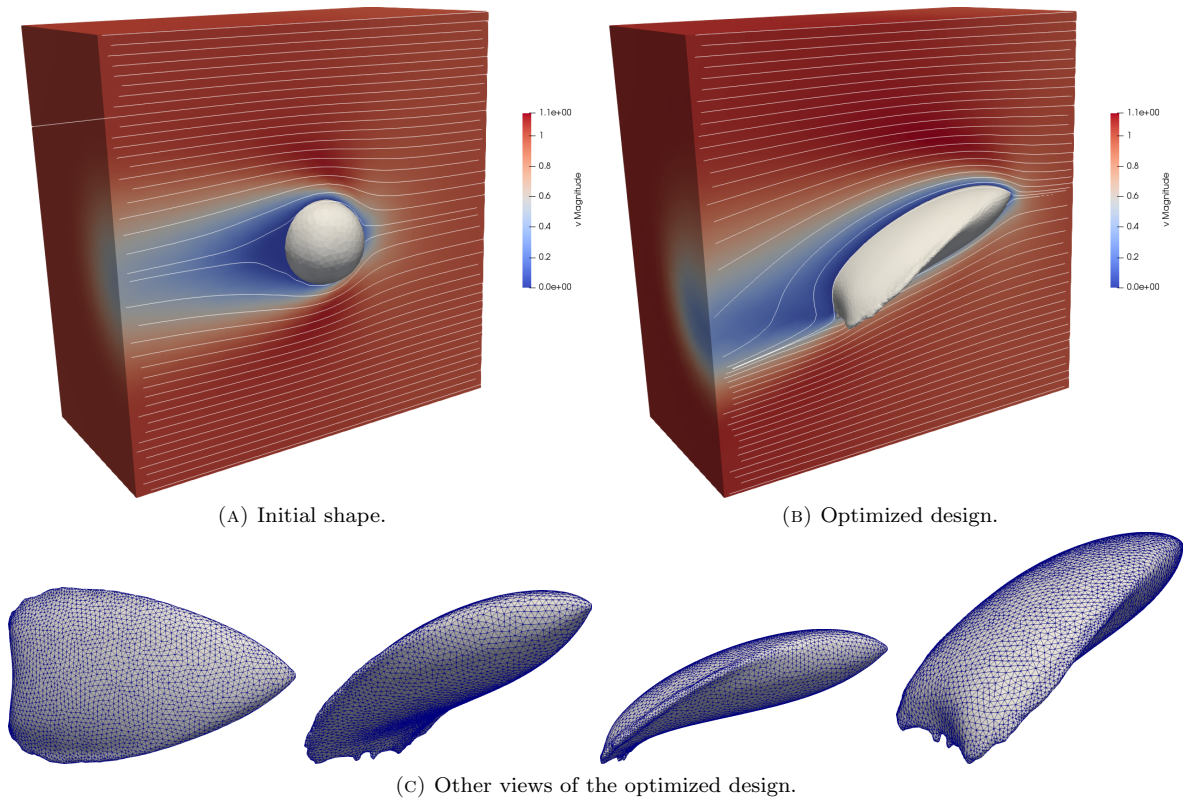
FIGURE 15. Optimized shape for the three-dimensional lift–drag maximization problem of section 5.3.

(A) Objective function Lift($\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)$).

(B) Volume constraint Vol(Ω_s)/Vol(D).

(C) Center of mass constraint $\mathbf{X}(\Omega_s) = (x, y, z)$.

(D) Drag constraint Drag($\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)$).
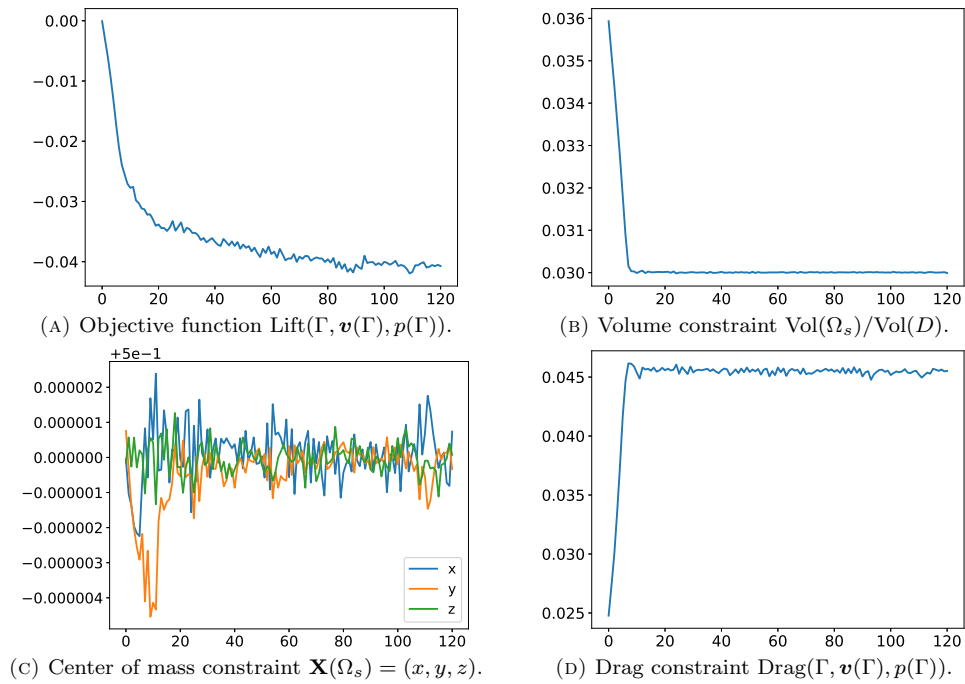
FIGURE 16. Convergence histories for the three-dimensional lift–drag optimization test case of section 5.3.
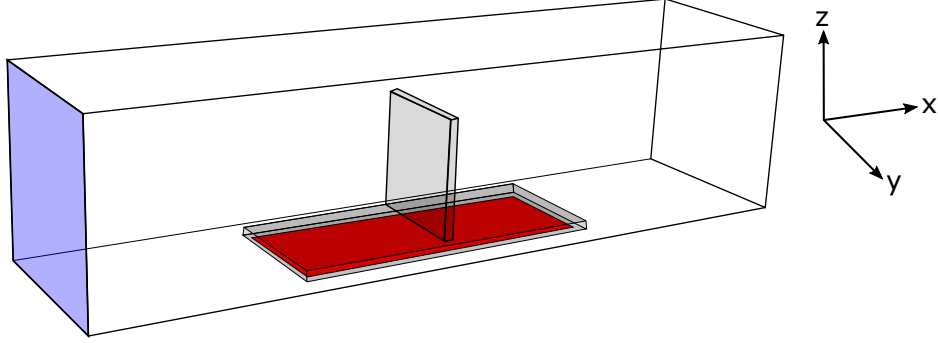
FIGURE 17. Setting of the fluid-structure interaction test case of section 5.4. A flow is entering from the left-hand side (in *blue*); a no-slip boundary condition is imposed at the bottom wall, and a slip boundary condition on the other side walls. The flow is pushing against a non optimizable vertical mechanical plate tightened to the bottom wall with a zero Dirichlet boundary condition (on the red surface).

Solving state equations	24 processes	15'
Computing shape derivatives and gradients	24 processes	2'
Advection of the level-set function	sequential	24"
Symmetrization and regularization of the level-set function	24 processes	13"
Remeshing	sequential	2'
Computation of the signed distance function	sequential	3'

TABLE 2. Runtimes for the first iteration of the fluid–structure interaction test case of section 5.4, using a mesh with 220,283 vertices.

The Reynolds number, fluid density, and viscosity are respectively set to Re = 60, $\rho = 1$, and $\nu = \rho H ||v_0||_\infty / \text{Re} = 0.012$. The characteristic length corresponding to the height of the non-optimizable plate is $H = 0.7$ and $||v_0||_\infty = 1$. The Lamé coefficients of the mechanical structure are $\lambda = 0.00529$ and $\mu = 0.0476$. The volume target equals 3 percent of that of the total domain: $V_T = 0.03\text{Vol}(D)$.

The optimized shape obtained is plotted on Figure 18. Not surprisingly, the final design has an aerodynamic profile in order to reduce the stress applied by the fluid flow.

For this test case, finite element computations were run in parallel on 24 processes. The number of mesh vertices for the first iteration is 220,283. This includes 132,775 nodes in the fluid domain and thus approximately 1.7 million degrees of freedom for the linearized fluid system. For the last iteration, there are 82,454 mesh nodes, 66,021 of which are in the fluid domain. Runtimes for the first mesh are listed in Table 2. The most CPU intensive task is the solution of the state equations. The runtimes mentioned include mesh decomposition steps, finite element matrix assemblies, nonlinear solver for the Navier–Stokes system, etc.

5.5. Shape and topology optimization of three-dimensional convective heat transfer

Our second and last multiphysics test case is concerned with convective heat transfer: the elastic deformation of the solid structure is not taken into account. This test case is the three-dimensional counterpart of a two-dimensional case considered e.g. in [74, 43].

The setting is depicted on Figure 22. The computational domain $D = [0,2] \times [0,1] \times [0,1]$ is a box divided into a fluid and a solid phase Ω_f, Ω_s, with different thermal conductivities. The fluid is entering end from the left side of D through a disk-shaped inlet boundary $\partial\Omega_f^{\text{in}}$ of radius 0.1. It is exiting at the right side through a disk-shaped outlet boundary $\partial\Omega_f^N$ of same radius. The input velocity profile v_0 is parabolic, with maximal velocity $||v_0||_\infty = 1$, and a (cold) temperature $T = 0$ is prescribed in this region. The flow is exiting
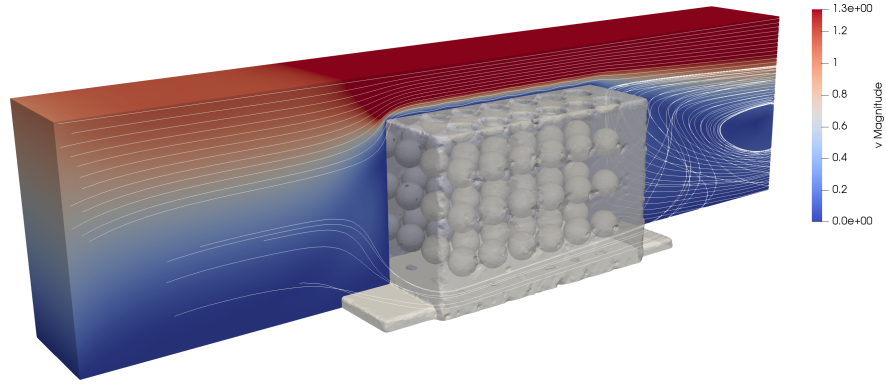
(A) Initial shape.
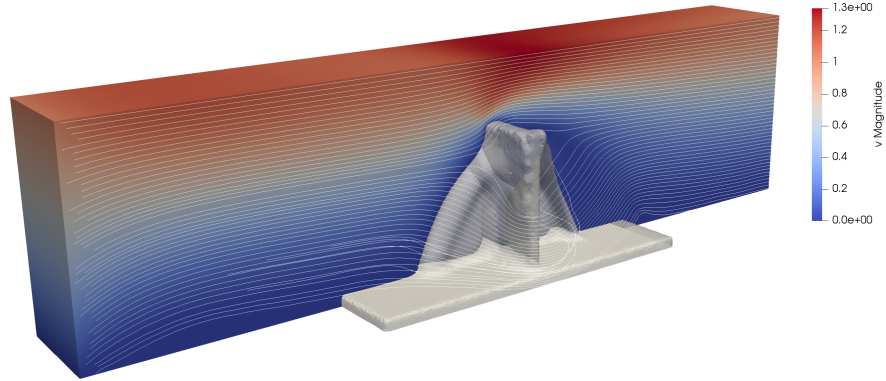
(B) Final design.
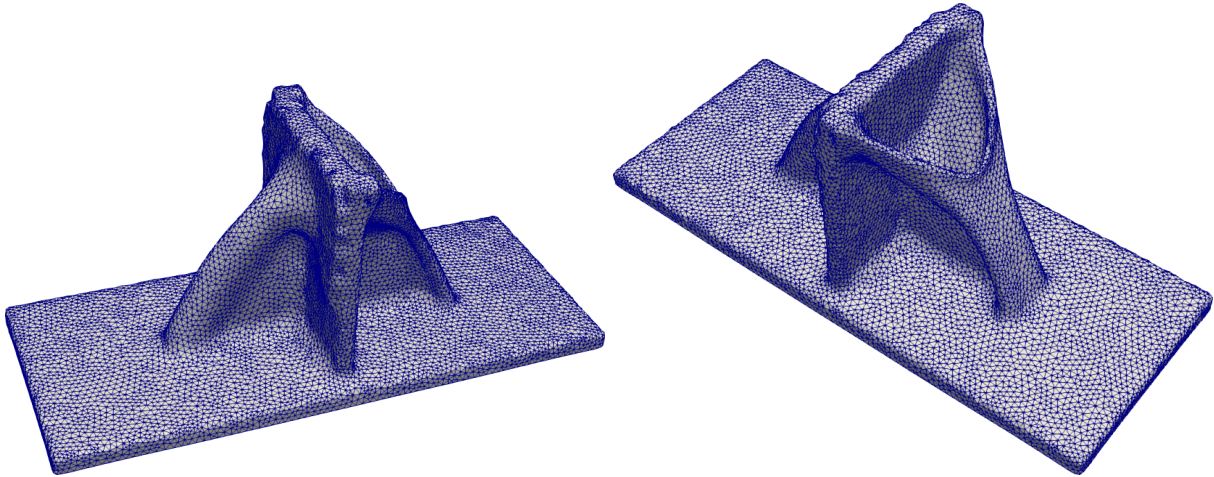
(C) Final design, front (resp. back) view on the left (resp. right).

FIGURE 18. Initial and optimized designs for the three-dimensional fluid–structure interaction test case of section 5.4.

from the right side of the domain through a disk-shaped outlet boundary bearing the Neumann boundary condition $\sigma_f(\boldsymbol{v}, p)\boldsymbol{n} = 0$. A stripe located at the middle of the outer boundary of the box D is kept at fixed (hot) temperature $T = 10$. The remaining regions of ∂D are considered adiabatic. We seek to optimize the shape of the solid–fluid interface Γ in order to maximize the heat stored by the fluid, under constraints on the static pressure loss and the volume occupied by the fluid. Mathematically, the following constrained
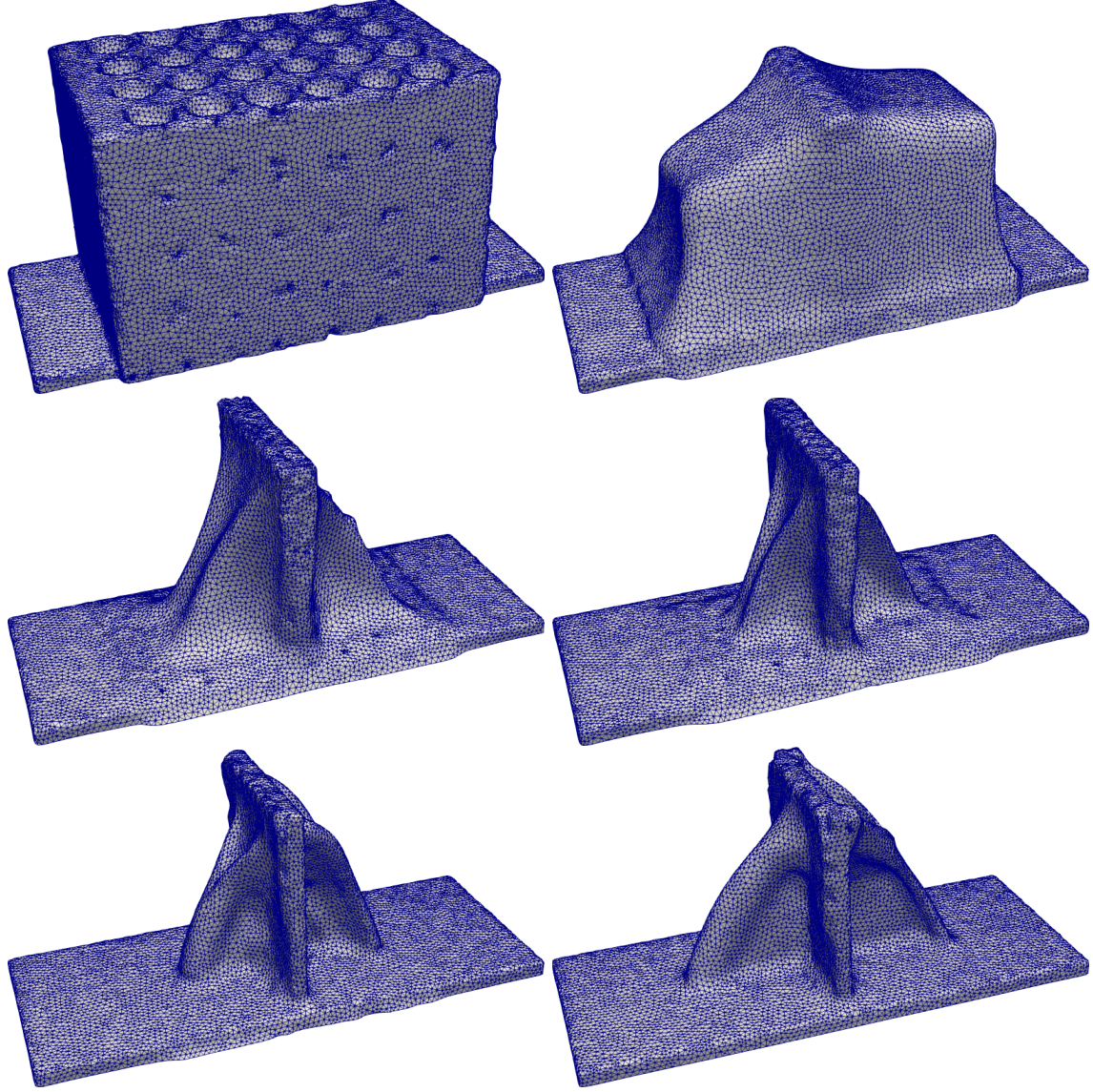
FIGURE 19. (From left to right, top to bottom) Intermediate iterations 0, 40, 100, 125, 175, and 300 for the fluid–structure test case of section 5.4.

optimization program is considered, see [43, 42]:

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

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

The fluid density is $\rho := 10$ and its viscosity is set to $\nu := 0.167$ which corresponds to a Reynolds number $\mathrm{Re} := \rho||\boldsymbol{v}_0||_\infty H/\nu = 60$ with a characteristic length $H = 1$. The fluid diffusivity and capacity coefficients are respectively set to $k_f := 1$ and $c_p := 500$ which corresponds to a Péclet number $\mathrm{Pe} := \rho c_p ||\boldsymbol{v}_0||_\infty H/k_f = 5,000$. The solid material is set to be 10 times more diffusive than the fluid material, i.e., $k_s := 10$.
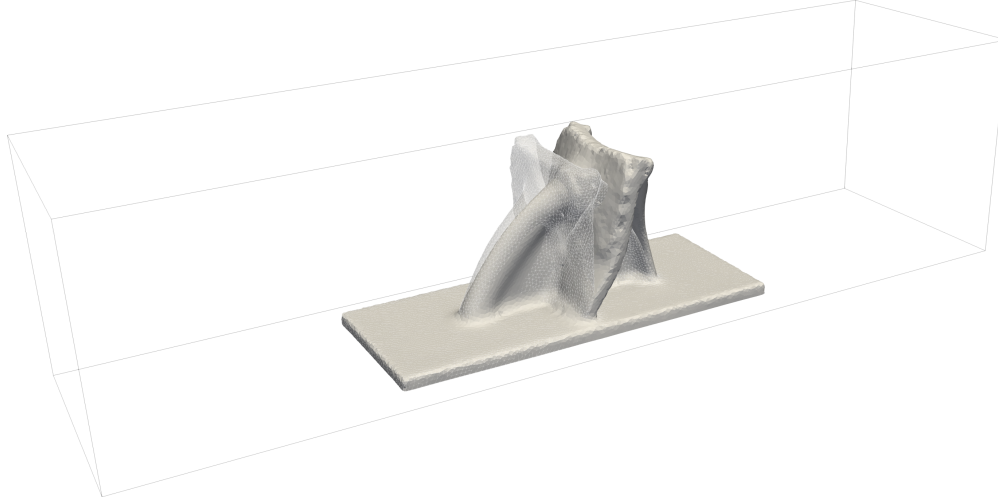
FIGURE 20. Linear elastic deformation of the optimized solid structure under the load force applied by the fluid for the test case of section 5.4.

(A) Objective function $J(\Omega_s, \boldsymbol{u}(\Omega_s))$.

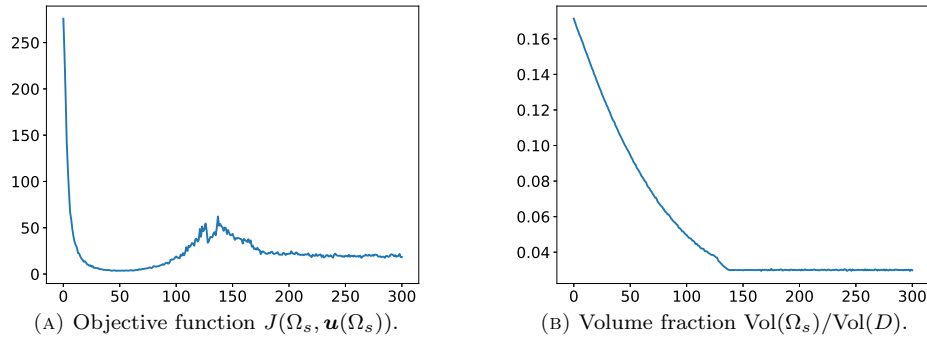(B) Volume fraction $\mathrm{Vol}(\Omega_s)/\mathrm{Vol}(D)$.

FIGURE 21. Convergence history for the fluid–structure interaction optimization test case of section 5.4.
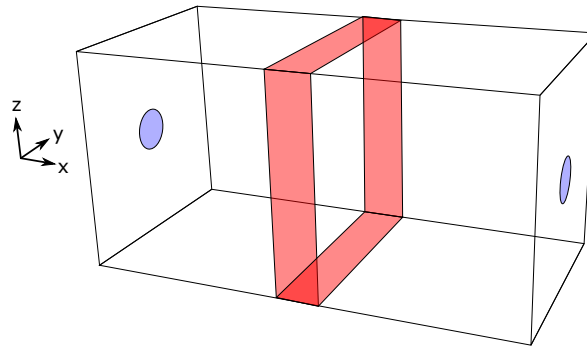
FIGURE 22. Setting for the convective heat transfer problem of section 5.5. A cold fluid is flowing from the left to the right blue disk-shaped boundaries and is heated by the red stripe side boundary over which a hot temperature is imposed.

Given these parameters, the upper bound threshold for the pressure drop is set to $\mathrm{DP}_T := 0.85$. The prescribed volume for the fluid phase is set to 20% of the total volume: $V_T = 0.2 \cdot \mathrm{Vol}(D)$.

The result of the optimization is shown on Figs. 24 and 25 and the associated convergence histories are reported in Figure 23. Intermediate iterations of the optimization process are plotted on Figure 26. The

27

initialization is a box filled with islands of solid spherical inclusions. Note that the final solid domain is connected although this requirement is not enforced explicitly. Similarly to what could be observed in the two-dimensional test case, the optimized design features thin inclusions of fluid attached to the main pipes so as to take advantage of the insulating low-diffusivity of the fluid.
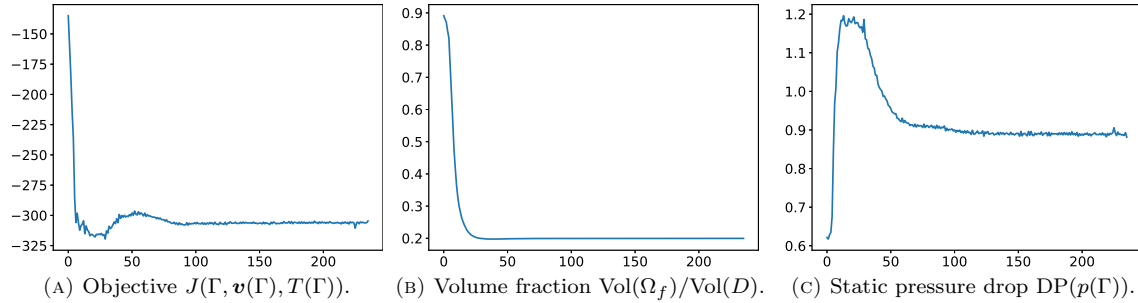

(A) Objective $J(\Gamma, \boldsymbol{v}(\Gamma), T(\Gamma))$. (B) Volume fraction $\mathrm{Vol}(\Omega_f)/\mathrm{Vol}(D)$. (C) Static pressure drop $\mathrm{DP}(p(\Gamma))$.

FIGURE 23. Convergence history for the convective heat transfer optimization test case of section 5.5.
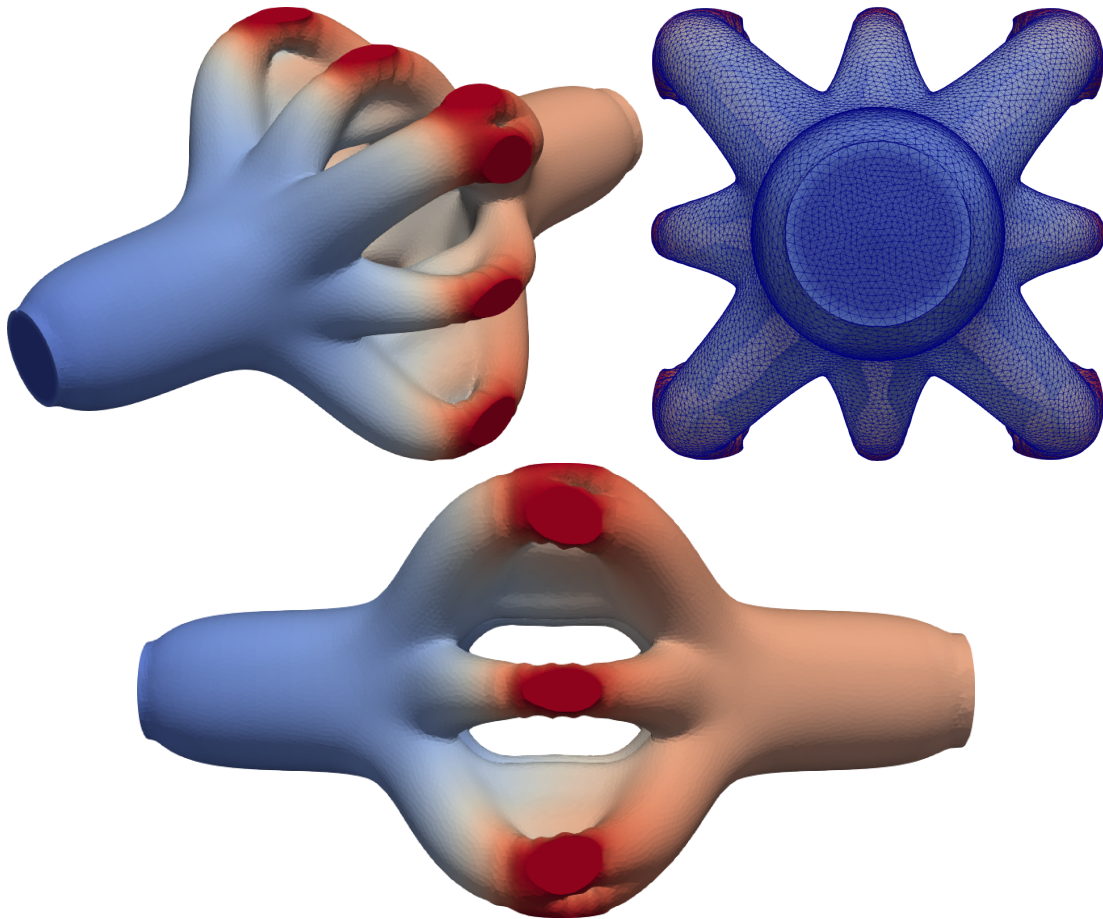

FIGURE 24. Optimized fluid domain for the convective heat transfer test case of section 5.5. The colors correspond to the temperature profile.

The meshes considered for this test case are characterized by a resolution ranging from approximately 267,000 to 616,800 vertices in the whole computational domain, and ranging from 126,000 to 300,000 vertices for the sole fluid domain. Hence, the resolution of the fluid system involves the inversion of finite element
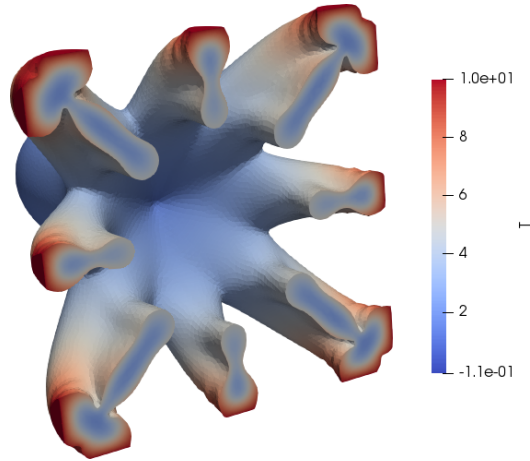
FIGURE 25. Sectional view of the optimized fluid domain for the convective heat transfer test case of section 5.5. The color corresponds to the temperature profile.

matrix with about 5×10^6 degrees of freedom. The whole optimization process takes approximately 3 days with the finite element related operations running in parallel on 24 processes.

REFERENCES

[1] N. AAGE, E. ANDREASSEN, AND B. S. LAZAROV, *Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 565–572.

[2] N. AAGE, E. ANDREASSEN, B. S. LAZAROV, AND O. SIGMUND, *Giga-voxel computational morphogenesis for structural design*, Nature, 550 (2017), p. 84.

[3] M. F. ADAMS, H. H. BAYRAKTAR, T. M. KEAVENY, AND P. PAPADOPOULOS, *Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom*, in Proceedings of the 2004 ACM/IEEE conference on Supercomputing, IEEE Computer Society, 2004, p. 34.

[4] S.-H. AHN AND S. CHO, *Level set–based topological shape optimization of heat conduction problems considering design-dependent convection boundary*, Numerical Heat Transfer, Part B: Fundamentals, 58 (2010), pp. 304–322.

[5] J. ALEXANDERSEN, O. SIGMUND, AND N. AAGE, *Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection*, International Journal of Heat and Mass Transfer, 100 (2016), pp. 876–891.

[6] G. ALLAIRE, *Conception optimale de structures*, vol. 58 of Mathématiques & Applications (Berlin), Springer-Verlag, Berlin, 2007.

[7] G. ALLAIRE, C. DAPOGNY, AND P. FREY, *Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh*, Comptes Rendus Mathematique, 349 (2011), pp. 999–1003.

[8] ———, *A mesh evolution algorithm based on the level set method for geometry and topology optimization*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 711–715.

[9] ———, *Shape optimization with a level set based mesh evolution method*, Computer Methods in Applied Mechanics and Engineering, 282 (2014), pp. 22–53.

[10] G. ALLAIRE AND L. JAKABCIN, *Taking into account thermal residual stresses in topology optimization of structures built by additive manufacturing*, Mathematical Models and Methods in Applied Sciences, 28 (2018), pp. 2313–2366.

[11] G. ALLAIRE, F. JOUVE, AND A. TOADER, *A level-set method for shape optimization*, Comptes Rendus Mathematique, 334 (2002), pp. 1125–1130.

[12] G. ALLAIRE, F. JOUVE, AND A.-M. TOADER, *Structural optimization using sensitivity analysis and a level-set method*, Journal of Computational Physics, 194 (2004), pp. 363–393.

[13] O. AMIR, N. AAGE, AND B. S. LAZAROV, *On multigrid-CG for efficient topology optimization*, Structural and Multidisciplinary Optimization, 49 (2014), pp. 815–829.
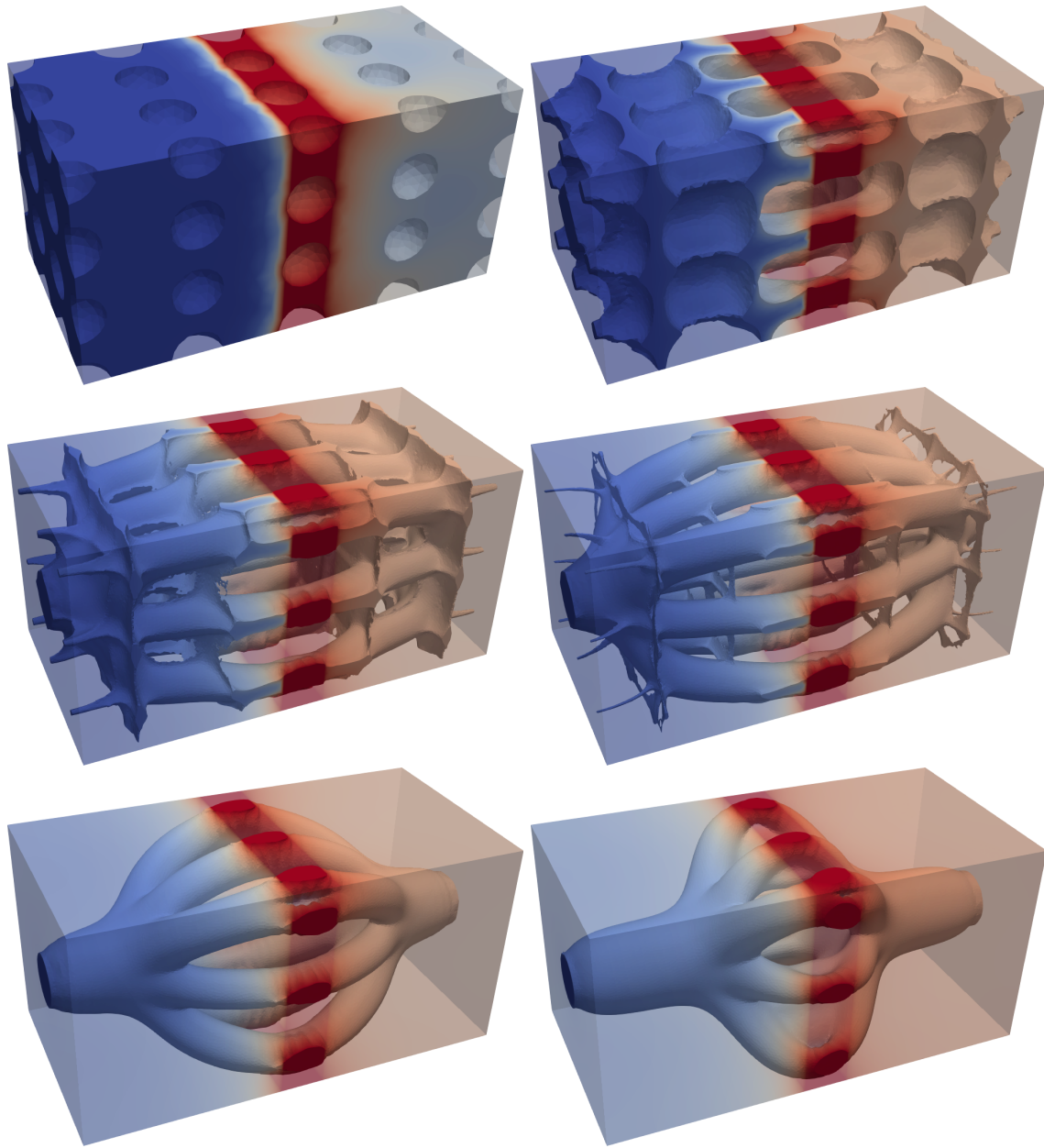
FIGURE 26. (From left to right, top to bottom) Intermediate iterations 0, 10, 15, 20, 50 and 235 for the convective heat transfer test case of section 5.5.

[14] C. S. ANDREASEN AND O. SIGMUND, *Topology optimization of fluid–structure-interaction problems in poroelasticity*, Computer Methods in Applied Mechanics and Engineering, 258 (2013), pp. 55–62.

[15] H. AZEGAMI, S. K. M. SHIMODA, AND E. KATAMINE, *Irregularity of shape optimization problems and an improvement technique*, WIT Transactions on The Built Environment, 31 (1997).

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

[17] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, A. DENER, V. EIJKHOUT, W. D. GROPP, D. KARPEYEV, D. KAUSHIK, M. G. KNEPLEY, D. A. MAY, L. C. MCINNES, R. T. MILLS, T. MUNSON, K. RUPP, P. SANAN, B. F. SMITH, S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 3.11, Argonne National Laboratory, 2019.

[18] ——, *PETSc Web page.* https://www.mcs.anl.gov/petsc, 2019.

[19] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.

[20] R. Becker, V. Heuveline, and R. Rannacher, *An optimal control approach to adaptivity in computational fluid mechanics*, International journal for numerical methods in fluids, 40 (2002), pp. 105–120.

[21] M. Benzi, M. A. Olshanskii, and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, International Journal for Numerical Methods in Fluids, 66 (2011), pp. 486–508.

[22] A. Bissuel, *Résolution des équations de Navier-Stokes linéarisées pour l'aéroélasticité, l?optimisation de forme et l?aéroacoustique*, PhD thesis, Paris Saclay, 2018.

[23] T. Borrvall and J. Petersson, *Large-scale topology optimization in 3D using parallel computing*, Computer methods in applied mechanics and engineering, 190 (2001), pp. 6201–6229.

[24] J. Céa, A. Gioan, and J. Michel, *Quelques résultats sur l'identification de domaines*, Calcolo. A Quarterly on Numerical Analysis and Theory of Computation, 10 (1973), pp. 207–232.

[25] J. Céa, A. Gioan, and J. Michel, *Adaptation de la méthode du gradient à un problème d'identification de domaine*, in Computing methods in applied sciences and engineering (Proc. Internat. Sympos., Versailles, 1973), Part 2, 1974, pp. 391–402. Lecture Notes in Comput. Sci., Vol. 11.

[26] C. Chevalier and F. Pellegrini, *PT-Scotch: a tool for efficient parallel graph ordering*, Parallel Computing, 34 (2008), pp. 318–331.

[27] C. P. Chukwudozie, *Shape Optimization for Drag Minimization Using the Navier-Stokes Equation*, Master's thesis, Louisiana State University, 2015.

[28] H. Chung, O. Amir, and H. A. Kim, *Nonlinear Thermoelastic Topology Optimization with the Level-Set Method*, in AIAA Scitech 2019 Forum, 2019, p. 1470.

[29] P. Coffin and K. Maute, *Level set topology optimization of cooling and heating devices using a simplified convection model*, Structural and Multidisciplinary Optimization, 53 (2016), pp. 985–1003.

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

[31] C. Dapogny, C. Dobrzynski, P. Frey, and A. Froelhy, *mmg, available at: https://www.mmgtools.org*, 2019.

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

[33] C. Dapogny, P. Frey, F. Omnès, and Y. Privat, *Geometrical shape optimization in fluid mechanics using FreeFem++*. HAL preprint hal-01481707, Mar. 2017.

[34] T. Dbouk, *A review about the engineering design of optimal heat transfer systems using topology optimization*, Applied Thermal Engineering, 112 (2017), pp. 841–854.

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

[36] E. M. Dede, *Multiphysics topology optimization of heat transfer and fluid flow systems*, in proceedings of the COMSOL Users Conference, 2009.

[37] S. B. Dilgen, C. B. Dilgen, D. R. Fuhrman, O. Sigmund, and B. S. Lazarov, *Density based topology optimization of turbulent flow heat transfer systems*, Structural and Multidisciplinary Optimization, 57 (2018), pp. 1905–1918.

[38] F. Dugast, Y. Favennec, C. Josset, Y. Fan, and L. Luo, *Topology optimization of thermal fluid flows with an adjoint Lattice Boltzmann Methodi*, Journal of Computational Physics, 365 (2018), pp. 376–404.

[39] T. Dupont, M. F. Wheeler, et al., *A Galerkin procedure for approximating the flux on the boundary for elliptic and parabolic boundary value problems*, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, 8 (1974), pp. 47–59.

[40] B. Epstein, A. Jameson, S. Peigin, D. Roman, N. Harrison, and J. Vassberg, *Comparative study of three-dimensional wing drag minimization by different optimization techniques*, Journal of Aircraft, 46 (2009), pp. 526–541.

[41] A. Faure, G. Michailidis, G. Parry, N. Vermaak, and R. Estevez, *Design of thermoelastic multi-material structures with graded interfaces using topology optimization*, Structural and Multidisciplinary Optimization, 56 (2017), pp. 823–837.

[42] F. Feppon, *Shape and topology optimization of multiphysics systems*, PhD thesis, Thèse de doctorat de l'Université Paris Saclay préparée à l'École polytechnique, 2019.

[43] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, and C. Dapogny, *Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework*, SeMA Journal, (2019), pp. 1–46.

[44] F. Feppon, G. Allaire, and C. Dapogny, *Null space gradient flows for constrained optimization with applications to shape optimization*, Submitted, (2019).

[45] P. J. Frey and P.-L. George, *Mesh generation: application to finite elements*, ISTE, 2007.

[46] H. Garcke, C. Hecht, M. Hinze, C. Kahle, and K. F. Lam, *Shape optimization for surface functionals in Navier-Stokes flow using a phase field approach*, Interfaces and Free Boundaries. Mathematical Analysis, Computation and Applications, 18 (2016), pp. 219–261.

[47] A. George, *Nested dissection of a regular finite element mesh*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 345–363.

[48] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, *OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization*, Structural and Multidisciplinary Optimization, 59 (2019), pp. 1075–1104.

[49] F. Hecht, *New development in FreeFem++*, Journal of Numerical Mathematics, 20 (2012), pp. 251–265.

[50] A. Henrot and M. Pierre, *Variation et optimisation de formes: une analyse géométrique*, vol. 48, Springer Science & Business Media, 2006.

[51] ———, *Shape variation and optimization*, vol. 28 of EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2018.

[52] A. Henrot and Y. Privat, *What is the optimal shape of a pipe?*, Archive for rational mechanics and analysis, 196 (2010), pp. 281–302.

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

[54] A. Jameson, *Aerodynamic design via control theory*, Journal of scientific computing, 3 (1988), pp. 233–260.

[55] ———, *Aerodynamic design via control theory*, in Recent advances in computational fluid dynamics (Princeton, NJ, 1988), vol. 43 of Lecture Notes in Engrg., Springer, Berlin, 1989, pp. 377–401.

[56] ———, *Computational algorithms for aerodynamic analysis and design*, Applied Numerical Mathematics. An IMACS Journal, 13 (1993), pp. 383–422.

[57] N. Jenkins and K. Maute, *Level set topology optimization of stationary fluid-structure interaction problems*, Structural and Multidisciplinary Optimization, 52 (2015), pp. 179–195.

[58] K. E. Jensen, *Solving stress and compliance constrained volume minimization using anisotropic mesh adaptation, the method of moving asymptotes and a global p-norm*, Structural and Multidisciplinary Optimization, 54 (2016), pp. 831–841.

[59] C. Jog, *Distributed-parameter optimization and topology design for non-linear thermoelasticity*, Computer Methods in Applied Mechanics and Engineering, 132 (1996), pp. 117–134.

[60] P. Jolivet, *FreeFEM tutorial*. http://jolivet.perso.enseeiht.fr/FreeFem-tutorial/, Apr 2019.

[61] S. Kambampati, C. Jauregui, K. Museth, and H. A. Kim, *Fast level set topology optimization using a hierarchical data structure*, in 2018 Multidisciplinary Analysis and Optimization Conference, 2018, p. 3881.

[62] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing, 20 (1998), pp. 359–392.

[63] T. Kondoh, T. Matsumori, and A. Kawamoto, *Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration*, Structural and Multidisciplinary Optimization, 45 (2012), pp. 693–701.

[64] A. Koshakji, A. Quarteroni, and G. Rozza, *Free form deformation techniques applied to 3D shape optimization problems*, Communications in Applied and Industrial Mathematics, 4 (2013), pp. e452, 26.

[65] S. Kreissl and K. Maute, *Levelset based fluid topology optimization using the extended finite element method*, Structural and Multidisciplinary Optimization, 46 (2012), pp. 311–326.

[66] S. Kreissl, G. Pingen, A. Evgrafov, and K. Maute, *Topology optimization of flexible micro-fluidic devices*, Structural and Multidisciplinary Optimization, 42 (2010), pp. 495–516.

[67] S. Kreissl, G. Pingen, and K. Maute, *Topology optimization for unsteady flow*, International Journal for Numerical Methods in Engineering, 87 (2011), pp. 1229–1253.

[68] A. B. Lambe and J. R. R. A. Martins, *Matrix-free aerostructural optimization of aircraft wings*, Structural and Multidisciplinary Optimization, 53 (2016), pp. 589–603.

[69] L. Leifsson and S. Koziel, *Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models: a review of recent progress*, Journal of Computational Science, 10 (2015), pp. 45–54.

[70] J. Li and S. Zhu, *Shape identification in stokes flow with distributed shape gradients*, Applied Mathematics Letters, 95 (2019), pp. 165–171.

[71] W. G. Litvinov, *On the optimal shape of a hydrofoil*, Journal of Optimization Theory and Applications, 85 (1995), pp. 325–345.

[72] C. Lundgaard, J. Alexandersen, M. Zhou, C. S. Andreasen, and O. Sigmund, *Revisiting density-based topology optimization for fluid-structure-interaction problems*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 969–995.

[73] G. Marck, M. Nemer, and J.-L. Harion, *Topology optimization of heat and mass transfer problems: Laminar flow*, Numerical Heat Transfer, Part B: Fundamentals, 63 (2013), pp. 508–539.

[74] G. Marck and Y. Privat, *On some shape and topology optimization problems in conductive and convective heat transfers*, in OPTI 2014, An International Conference on Engineering and Applied Sciences Optimization, M. Papadrakakis, M. Karlaftis, and N. Lagaros, eds., June 2014, pp. 1640–1657.

[75] J. Martínez-Frutos and D. Herrero-Pérez, *GPU acceleration for evolutionary topology optimization of continuum structures using isosurfaces*, Computers & Structures, 182 (2017), pp. 119–136.

[76] L. A. M. Mello, E. De Sturler, G. H. Paulino, and E. C. N. Silva, *Recycling Krylov subspaces for efficient large-scale electrical impedance tomography*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 3101–3110.

[77] W. D. Middleton, *A numerical method for the design of camber surfaces of supersonic wings with arbitrary planforms*, tech. rep., NASA Technical report, 1964.

[78] B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, second ed., 2010.

[79] J. Moulin, P. Jolivet, and O. Marquet, *Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis*, Computer Methods in Applied Mechanics and Engineering, 351 (2019), pp. 718–743.

[80] D. J. MUNK, T. KIPOUROS, G. A. VIO, G. T. PARKS, AND G. P. STEVEN, *On the effect of fluid-structure interactions and choice of algorithm in multi-physics topology optimisation*, Finite Elements in Analysis and Design, 145 (2018), pp. 32–54.

[81] F. MURAT AND J. SIMON, *Sur le contrôle par un domaine géométrique*, Publication du Laboratoire d'Analyse Numérique de l'Université Pierre et Marie Curie, (1976).

[82] H. OKUMURA AND M. KAWAHARA, *Shape optimization of body located in incompressible Navier-Stokes flow based on optimal control theory*, CMES. Computer Modeling in Engineering & Sciences, 1 (2000), pp. 71–77.

[83] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.

[84] C. PECHSTEIN, *Preliminaries*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–61.

[85] P.-O. PERSSON, *Mesh generation for implicit geometries*, PhD thesis, Massachusetts Institute of Technology, 2005.

[86] M. PIETROPAOLI, F. MONTOMOLI, AND A. GAYMANN, *Three-dimensional fluid topology optimization for heat transfer*, Structural and Multidisciplinary Optimization, 59 (2019), pp. 801–812.

[87] O. PIRONNEAU, *On optimum profiles in Stokes flow*, Journal of Fluid Mechanics, 59 (1973), pp. 117–128.

[88] ——, *On optimum design in fluid mechanics*, Journal of Fluid Mechanics, 64 (1974), pp. 97–110.

[89] ——, *Optimal shape design for elliptic systems*, in System Modeling and Optimization, Springer, 1982, pp. 42–66.

[90] P. I. PLOTNIKOV AND J. SOKOLOWSKI, *Optimal shape control of airfoil in compressible gas flow governed by Navier-Stokes equations*, Evolution Equations and Control Theory, 2 (2013), pp. 495–516.

[91] N. POLLINI, O. SIGMUND, C. S. ANDREASEN, AND J. ALEXANDERSEN, *A "poor man's" approach for high-resolution three-dimensional topology design for natural convection problems*, Advances in Engineering Software, 140 (2020), p. 102736.

[92] J. QIAN, Y.-T. ZHANG, AND H.-K. ZHAO, *Fast sweeping methods for eikonal equations on triangular meshes*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 83–107.

[93] Y. SAAD, *A Flexible Inner–Outer Preconditioned GMRES Algorithm*, SIAM Journal on Scientific Computing, 14 (1993), pp. 461–469.

[94] Y. SAAD, *Iterative methods for sparse linear systems*, vol. 82, siam, 2003.

[95] Y. SAAD AND M. SCHULTZ, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[96] M. A. SALAZAR DE TROYA AND D. A. TORTORELLI, *Adaptive mesh refinement in stress-constrained topology optimization*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 2369–2386.

[97] K. R. SAVIERS, R. RANJAN, AND R. MAHMOUDI, *Design and validation of topology optimized heat exchangers*, in AIAA Scitech 2019 Forum, 2019, p. 1465.

[98] S. SCHMIDT AND V. SCHULZ, *A 2589 line topology optimization code written for the graphics card*, Computing and Visualization in Science, 14 (2011), pp. 249–256.

[99] J. A. SETHIAN, *Fast marching methods*, SIAM review, 41 (1999), pp. 199–235.

[100] O. SIGMUND AND S. TORQUATO, *Design of materials with extreme thermal expansion using a three-phase topology optimization method*, Journal of the Mechanics and Physics of Solids, 45 (1997), pp. 1037–1067.

[101] J. SOKOLOWSKI AND J.-P. ZOLÉSIO, *Introduction to shape optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.

[102] M. SONNTAG, S. SCHMIDT, AND N. R. GAUGER, *Shape derivatives for the compressible Navier-Stokes equations in variational form*, Journal of Computational and Applied Mathematics, 296 (2016), pp. 334–351.

[103] M. G. UKKEN AND M. SIVAPRAGASAM, *Aerodynamic shape optimization of airfoils at ultra-low Reynolds numbers*, Sādhanā, 44 (2019), p. 130.

[104] M. Y. WANG, X. WANG, AND D. GUO, *A level set method for structural topology optimization*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.

[105] O. WIDLUND AND M. DRYJA, *An additive variant of the Schwarz alternating method for the case of many subregions*, Technical Report 339, Ultracomputer Note 131, Department of Computer Science, Courant Institute, 12 1987.

[106] Q. XIA AND M. Y. WANG, *Topology optimization of thermoelastic structures using level set method*, Computational Mechanics, 42 (2008), pp. 837–857.

[107] K. YAJI, T. YAMADA, S. KUBO, K. IZUI, AND S. NISHIWAKI, *A topology optimization method for a coupled thermal–fluid problem using level set boundary expressions*, International Journal of Heat and Mass Transfer, 81 (2015), pp. 878–888.

[108] G. H. YOON, *Stress-based topology optimization method for steady-state fluid–structure interaction problems*, Computer Methods in Applied Mechanics and Engineering, 278 (2014), pp. 499 – 523.

[109] X. ZHAO, M. ZHOU, O. SIGMUND, AND C. ANDREASEN, *A "poor man's approach" to topology optimization of cooling channels based on a Darcy flow model*, International Journal of Heat and Mass Transfer, 116 (2018), pp. 1108–1123.

[110] S. ZHU, *Effective shape optimization of laplace eigenvalue problems using domain expressions of eulerian derivatives*, Journal of Optimization Theory and Applications, 176 (2018), pp. 17–34.