GEOMETRICAL SHAPE OPTIMIZATION IN FLUID MECHANICS USING FREEFEM++

CHARLES DAPOGNY[1], PASCAL FREY [2,3], FLORIAN OMNÈS [2], YANNICK PRIVAT [2]

[1] *Laboratoire Jean Kuntzmann, CNRS, Université Grenoble-Alpes, BP 53, 38041 Grenoble Cedex 9, France,*
[2] *Sorbonne Universités, UPMC Univ Paris 06, CNRS UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France,*
[3] *Sorbonne Universités, UPMC Univ Paris 06, Institut des Sciences du Calcul et des Données (ISCD), F-75005, Paris, France*

Abstract. In this article, we present simple and robust numerical methods for two-dimensional geometrical shape optimization problems, in the context of viscous flows driven by the stationary Navier-Stokes equations at low Reynolds number. The salient features of our algorithm are exposed with an educational purpose; in particular, the numerical resolution of the nonlinear stationary Navier-Stokes system, the Hadamard boundary variation method for calculating the sensitivity of the minimized function of the domain, and the mesh update strategy are carefully described. Several pedagogical examples are discussed. The corresponding programs are written in the `FreeFem++` environment; they are freely available and can easily be elaborated upon to deal with different, or more complex physical situations.

Contents

---

1. Introduction

The first industrial developments of shape optimization in situations involving fluid mechanics arose in the fields of aeronautic and aerospatial engineering. These developments were motivated by the tremendous production and running costs of aircraft: even small improvements on the performance of a design entail very large savings. Perhaps the most famous issue in this field is the design optimization of an airfoil, which dates back to at least 1964 [17]; see also [40, 41], and [55] where optimal profiles for minimum drag problems are calculated thanks to shape sensitivity analyses. We generally refer to [35], Chap. 1 for a historical perspective about the emergence of optimal design techniques in the context of fluid mechanics. Since the aforementioned pioneering works, applications of shape optimization in fluid mechanics have raised a great interest in various areas such as the automotive industry - see [19] about the numerical optimization of a cooling fan - or in computational biology: for instance, in [2, 3], the design optimization of an artery graft for preventing the formation of a stenosis is investigated from a numerical point of view.

Let us briefly outline some of the shape optimization strategies carried out in the literature, without looking for exhaustivity. For more in-depth discussions in the context of fluid mechanics, we refer to [35, 46], or to the review article [45]. Any shape optimization method relies on a parametrization of shapes, that is, on the definition of a set of design variables. Depending on the context, these design variables may be physical parameters of shapes (the length, or thickness of some parts), control points of a CAD description, or the vertices of a meshed representation. In such a context, the sensitivity (i.e. derivative) of the objective and constraint functionals of the optimization problem with respect to the design variables - which is a key ingredient in most numerical optimization algorithms - can be evaluated either by approximate methods (for instance by finite differences featuring small perturbations of the parameters), or analytically, by relying on adjoint techniques from optimal control theory [42, 43, 56]. In this last class of methods, which is by now quite popular, these sensitivities may be calculated at the discrete level (i.e. the derivative of the finite-dimensional functional resulting from the discretization of the shape is considered), which requires a perfect knowledge of the discretization and numerical resolution of the flow equations (but allows for the use of automatic differentiation methods), or at the continuous level, which relies on advanced mathematical tools, but completely decouples the calculation of the derivatives and the numerical resolution of the problem.

In any event, a great numerical challenge faced by all these methods is that of updating the design of the shape from one iteration of the process to the next, while avoiding that the numerical representation becomes invalid. Recently, several strategies have been thought off to circumvent this difficulty, and more generally to allow for more freedom in terms of the variety of designs that can be represented, to the point that they make it possible to account for changes in their topology.

In this direction, quite popular density-based methods in structural mechanics - and notably the famous SIMP method (see [10] and references therein) - have been introduced in the context of fluid mechanics in [13]; see also [54] and [1], where a large-scale example is discussed. These relaxation methods rely on an extension of the set of admissible designs: 'black-and-white' shapes Ω, contained in a fixed computational domain D, or equivalently their characteristic function $\chi : D \to \{0, 1\}$, taking values 1 inside Ω, and 0 in the 'void' region $D \setminus \overline{\Omega}$, are replaced with density functions $\rho : D \to [0, 1]$, which may assume intermediate, 'grayscale' values in $(0, 1)$. The flow equations have then to be given an appropriate meaning to account for the presence of 'void' and 'grayscale' regions. This is typically achieved by adding a ρ-dependent damping term (or Brinkman's law) to the flow equations [13], a heuristic inspired from the theory of porous media whereby the void is filled with a fluid with very low permeability, simulating no slip boundary conditions at the interface between the fluid and void domains (see [34] and [29], then [32] for a generalization to the case of Navier-Stokes flows).

Another class of shape optimization strategies relies on the level set method, pioneered in [52], then introduced in structural optimization in [57, 5, 61]. Such methods describe a shape as the negative sub-domain of a scalar 'level set' function defined on the computational domain D. In the two-dimensional work [27], the level set method is used to deal with Navier-Stokes flows, in a variational framework which alleviates the need for the redistancing stage inherent to many level set based algorithms; this idea is continued in [62] in the three-dimensional setting. See also [14] for another use of the Level Set method in the context of Navier-Stokes flows. In [18], the Level Set method is used to combine the information supplied by shape and topological derivatives, in the context of Stokes flows, in two and three space dimensions.

Eventually, let us also mention phase-field methods, which share a lot of features with the level set method, except for the fact that they feature shapes, or phases, with 'thickened boundaries' [31].

Following the lead of [58] and [6], which take place in the context of structural mechanics, this article is a pedagogical introduction to several basic shape optimization techniques in fluid mechanics. The discussion is didactic: we deliberately keep technicalities to a minimum, and provide adequate references when needed. We present a simple numerical framework, yet robust enough to deal with physically relevant situations, which is dedicated to solving shape optimization problems in the context of fluid mechanics. The proposed examples can be easily reproduced and elaborated upon to deal with more advanced models.

In the setting of the stationary Navier-Stokes equations at low Reynolds number, we optimize shapes in terms e.g. of the dissipated viscous energy, under volume or perimeter constraints. To this end, we rely on an augmented Lagrangian algorithm based on the first order information supplied by shape derivatives, in the sense of the Hadamard boundary variation method. From the numerical point of view, shapes are represented via a computational mesh, on which the flow equations are solved. The update of the shape between each iteration of the optimization process is achieved by moving the vertices of this mesh according to the calculated descent direction.

The numerical developments proposed in this article rely on the `FreeFem++` [36] software, a free environment allowing to solve a wide variety of Partial Differential Equations (PDE for short) using the Finite Element method within a few command lines.

A particular attention has been paid to the development of a user-friendly source code, which is available online at

`https://github.com/flomnes/optiflow`

with the hope that it serve as a useful basis for further investigations.

The remainder of this article is organized as follows. In Section 2, we introduce the model physical problem at stake, as well as the shape optimization problem considered in this context. In passing, we recall in an elementary way some basic facts about shape derivatives. In Section 3, we describe in more details the main ingredients of the proposed numerical method: after a short motivating outline in Section 3.1, we discuss the salient features of our shape optimization algorithm in Sections 3.2, 3.3, 3.4, 3.5 and 3.6; a sketch of this algorithm is then provided in Section 3.7. In Section 4, we introduce and comment five test cases which are dealt with by our algorithm. Finally, Section 5 concludes by evoking limitations of our approach as well as perspectives for possible improvements and extensions.

2. Shape optimization for flows governed by the Navier-Stokes equations

In this section, we present the model physical situation and the shape optimization problem at stake, together with the necessary theoretical background. Notice that, while the concrete applications discussed in this article arise in two space dimensions (see Section 4), most of the presented techniques are available in the general, d-dimensional setting. For this reason, the discussion takes place in d dimensions inasmuch as it is possible without jeopardizing with clarity.

2.1. **The Navier-Stokes equations.**

In our applications, shapes are smooth bounded domains $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, occupied by a homogeneous Newtonian fluid with kinematic viscosity $\nu > 0$. A velocity profile $\mathbf{u}_{\text{in}}$ is imposed on a subset Γ_{in} of $\partial\Omega$ (the inlet), and another, disjoint subset $\Gamma_{\text{out}} \subset \partial\Omega$ (the outlet) is left free of surface forces. No slip boundary conditions are imposed on the remaining, 'free' part $\Gamma := \partial\Omega \setminus \overline{(\Gamma_{\text{in}} \cup \Gamma_{\text{out}})}$ of the boundary; see Figure 1 for an illustration. From the physical point of view, Ω may represent a mammal's lung, a duct in a ventilation system or a water radiator, etc.

The equilibrium behavior of the fluid inside Ω is classically described in terms of its (vector) velocity field $\mathbf{u} = (u_1, ..., u_d) : \Omega \to \mathbb{R}^d$ and (scalar) pressure $p : \Omega \to \mathbb{R}$, which solve the stationary *incompressible*

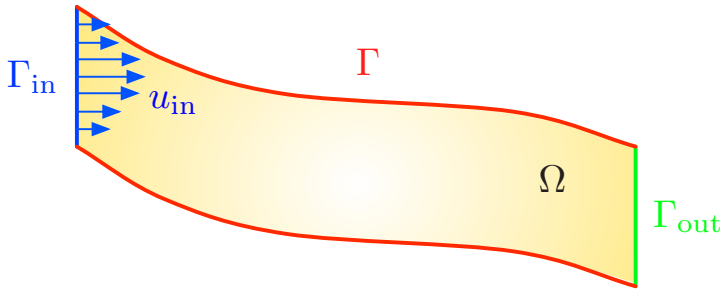

FIGURE 1. *Illustration of the model setting introduced in Section 2.*

Navier-Stokes equations:

$$\text{(2.1)} \qquad \begin{cases} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = 0 & \text{in } \Omega, \\ \operatorname{div}(\mathbf{u}) = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{u}_{\text{in}} & \text{on } \Gamma_{\text{in}}, \\ \mathbf{u} = 0 & \text{on } \Gamma, \\ \sigma(\mathbf{u},p)\mathbf{n} = 0 & \text{on } \Gamma_{\text{out}}. \end{cases}$$

In the above equation, the stress tensor $\sigma(\mathbf{u},p)$ is defined by

$$\sigma(\mathbf{u},p) = 2\nu e(\mathbf{u}) - p\mathrm{I}, \text{ where } e(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u}^T + \nabla\mathbf{u}).$$

For further reference, let us recall that this nonlinear system is often considered from the variational viewpoint, in particular when it comes to its numerical resolution; the pair $(\mathbf{u},p)$ satisfies:

$$\text{(2.2)} \qquad \begin{cases} a(\mathbf{u},\mathbf{v}) + c(\mathbf{u},\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) &= 0 \\ b(\mathbf{u},q) &= 0 \end{cases} \quad \text{for all } (\mathbf{v},q) \text{ s.t. } \mathbf{v} = 0 \text{ on } \Gamma_{in},$$

where we have defined

$$a(\mathbf{u},\mathbf{v}) = 2\nu\int_\Omega e(\mathbf{u}) : e(\mathbf{v})\,dx, \quad c(\mathbf{u},\mathbf{v},\mathbf{w}) = \int_\Omega (\mathbf{u}\cdot\nabla)\mathbf{v}\cdot\mathbf{w}\,dx, \quad b(\mathbf{u},p) = -\int_\Omega p\operatorname{div}(\mathbf{u})\,dx,$$

and the notation $A : B$ stands for the usual Frobenius inner product of two $d\times d$ matrices A, B, that is $A : B = \sum_{i,j=1}^d A_{ij}B_{ij}$. In the following, we also denote by $||A|| = (A : A)^{1/2}$ the associated Frobenius norm.

In the dimensionless version (2.1), (2.2) of the Navier-Stokes equations, the Reynolds number Re is proportional to $1/\nu$: it is an indicator of the type of regime of the flow [47]. At low Reynolds number, viscous effects are prevailing and the flow is laminar; in particular, its velocity stays relatively low. On the contrary, at moderate to high Reynolds number, convective forces become dominant and the flow is turbulent. The theoretical and numerical study of (2.1) is notoriously much harder in the latter situation, and still leaves room for many open questions. In the present, introductory work, we limit ourselves to the low Reynolds number regime.

Remark 1. Let us say a few words about the functional setting and well-posedness of the stationary Navier-Stokes system (2.1). When the viscosity ν is large enough, i.e. the Reynolds number Re is low, (2.1) is well-posed. It has a unique weak solution $(\mathbf{u},p) \in H^1(\Omega)^d \times L^2_0(\Omega)$, where $L^2_0(\Omega) := \left\{p \in L^2(\Omega),\ \int_\Omega p\,dx = 0\right\}$, in the sense that the variational problem (2.2) is fulfilled; see [60], Chapter II about these matters. In the following, we systematically assume ν to be large enough so that (2.1) is well-posed.

2.2. **Presentation of the shape optimization problem.**

In the context of Section 2.1, the shape optimization problem of interest reads

$$\text{(2.3)} \qquad \text{minimize } J(\Omega) \text{ over the set } \mathcal{O}_{ad} \text{ under the constraint } G(\Omega) = 0.$$

Here, the objective criterion $J(\Omega)$ may stand for

- The energy $E(\Omega)$ dissipated by the fluid owing to the work of viscous forces, i.e.

$$E(\Omega) = \int_\Omega \sigma(\mathbf{u}, p) : e(\mathbf{u})\, dx = 2\nu \int_\Omega \|e(\mathbf{u})\|^2\, dx, \tag{2.4}$$

- A least-square discrepancy

$$D(\Omega) = \frac{1}{2} \int_{\Gamma_{\text{out}}} |\mathbf{u} - \mathbf{u}_{\text{ref}}|^2\, ds \tag{2.5}$$

between the velocity $\mathbf{u}$ of the fluid, solution to (2.1), and a given reference profile $\mathbf{u}_{\text{ref}}$. Such criteria are often involved in shape optimization-based methods for the detection or the reconstruction of an obstacle immersed in a fluid from the data of boundary measurements [8, 44].

All the considered domains enclose the inlet Γ_{in} and the outlet Γ_{out} as (fixed) subsets of their boundaries, so that the free boundary Γ is the only region of $\partial\Omega$ subject to optimization. Accordingly, the set $\mathcal{O}_{ad}$ of admissible domains featured in (2.3) reads:

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^d, \text{ open, smooth and bounded, such that } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \subset \partial\Omega\} \tag{2.6}$$

Last but not least, as far as the constraint functional $G(\Omega)$ is concerned, we shall restrict ourselves to equality constraints on the volume or the perimeter of shapes, namely:

$$G(\Omega) = \text{Vol}(\Omega) - V_T, \text{ or } G(\Omega) = \text{Per}(\Omega) - P_T$$

for some given volume or perimeter target values V_T and P_T.

Remark 2. The existence of solutions to problems of the form (2.3) is a long-standing question in shape optimization theory, not only in the context of fluid mechanics, but already in simpler situations, bringing into play the conductivity equation, or the linearized elasticity system. Let us simply mention that, in order to guarantee the existence of optimal shapes, two classical remedies consist in either restricting the set of admissible shapes (for instance by adding constraints on the perimeter, or the regularity of shapes), or on the contrary in enlarging this set, so that it includes 'density functions', and not only 'black and white' shapes. See for instance [15, 37, 59] about these issues, or [38, 39, 12] in the context of fluid mechanics.

2.3. **Shape sensitivity analysis using Hadamard's boundary variation method.**

Most optimization algorithms - such as steepest-descent methods, as used in our applications - rely on the knowledge of the derivatives of the objective and constraint functionals. As we have already hinted at in the introduction, two different paradigms exist in the context of PDE constrained optimization problems of the form (2.3). In a nutshell, in 'discretize-then-optimize' approaches, the optimized domain is first discretized into a set of design variables (for instance, the vertices of a mesh); the PDE system (2.1) becomes finite-dimensional (it is e.g. discretized using a Finite Element method), and its coefficients depend on the design variables; accordingly, the objective and constraint functionals $J(\Omega)$ and $G(\Omega)$ are functions of the design variables, and the derivatives of these discrete functionals are calculated. On the contrary, 'optimize-then-discretize' approaches advocate to calculate the derivatives of $J(\Omega)$ and $G(\Omega)$ at the continuous level; the resulting theoretical formulae are then discretized by relying on a discretization of the domain and of the PDE system (2.1).

Our approach belongs to the last category, and therefore requires to perform derivatives with respect to the domain. Several ways exist to define a notion of shape sensitivity, and we presently rely on Hadamard's boundary variation method, referring for instance to [37], Chap. 5, or [4, 49] for in-depth expositions. See also [51] for an overview of the rival notion of *topological derivative*, and [7] for the calculation of topological derivatives in the context of fluid mechanics.

In the framework of Hadamard's method, the sensitivity of a function of the domain is appraised with respect to small perturbations of its boundary: variations of a given shape Ω are considered in the form

$$\Omega_{\boldsymbol{\theta}} = (\text{Id} + \boldsymbol{\theta})(\Omega), \text{ where } \boldsymbol{\theta} : \mathbb{R}^d \to \mathbb{R}^d \text{ is a 'small' vector field,} \tag{2.7}$$

and Id is the identity mapping from $\mathbb{R}^d$ into itself; see Figure 2 for an illustration.

Since admissible shapes $\Omega \in \mathcal{O}_{ad}$ are smooth and only Γ is subject to optimization, it is natural that $\boldsymbol{\theta}$ belong to the set Θ_{ad} of admissible perturbations defined by:

$$\Theta_{ad} = \left\{ \boldsymbol{\theta} : \mathbb{R}^d \to \mathbb{R}^d \text{ smooth}, \ \boldsymbol{\theta} = 0 \text{ on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \right\};$$

doing so guarantees that variations (2.7) of admissible shapes stay admissible.

Definition 1. *A function of the domain $F(\Omega)$ is shape differentiable at $\Omega \in \mathcal{O}_{ad}$ if the underlying mapping $\boldsymbol{\theta} \mapsto F(\Omega_{\boldsymbol{\theta}})$, from Θ_{ad} into $\mathbb{R}$, is differentiable at $\boldsymbol{\theta} = 0$ (in the sense of Fréchet). The corresponding derivative is denoted by $\boldsymbol{\theta} \mapsto F'(\Omega)(\boldsymbol{\theta})$, and the following Taylor expansion holds:*

$$F(\Omega_{\boldsymbol{\theta}}) = F(\Omega) + F'(\Omega)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \ \text{where } o(\boldsymbol{\theta}) \to 0 \text{ as } \boldsymbol{\theta} \to 0. \tag{2.8}$$

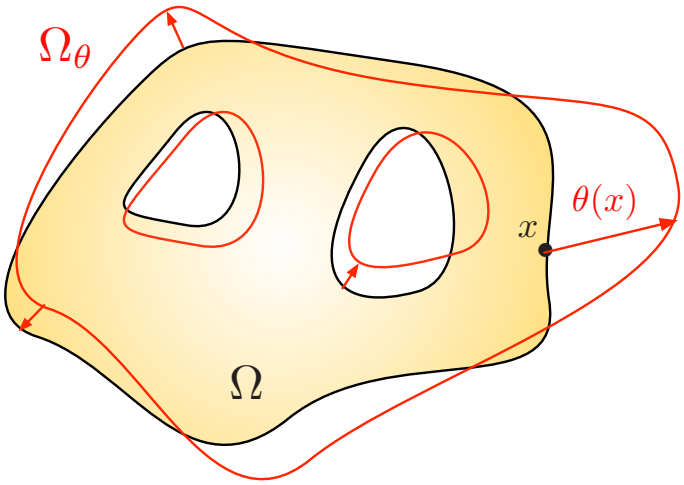

FIGURE 2. *Example of a variation $\Omega_{\boldsymbol{\theta}}$ of a shape Ω.*

When it comes to shape derivatives, the first result of interest deals with the volume and perimeter functionals; see [4, 37] for a proof.

Theorem 1. *Let Ω be a smooth shape. Then,*

(i) The volume $\mathrm{Vol}(\Omega)$ *is shape differentiable and its derivative reads:*

$$\forall \boldsymbol{\theta} \in \Theta_{ad}, \ \mathrm{Vol}'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta} \cdot \mathbf{n} \, ds.$$

(ii) The perimeter $\mathrm{Per}(\Omega)$ *is shape differentiable and its derivative reads:*

$$\forall \boldsymbol{\theta} \in \Theta_{ad}, \ \mathrm{Per}'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \kappa \, \boldsymbol{\theta} \cdot \mathbf{n} \, ds,$$

where $\kappa : \partial\Omega \to \mathbb{R}$ is the mean curvature of $\partial\Omega$.

Calculating shape derivatives of functions of the form (2.4) or (2.5) is a little harder, since they bring into play the solution of a partial differential equation posed on Ω (in the present case, the Navier-Stokes system (2.1)). This can however be managed by using quite classical adjoint techniques from optimal control theory. Again, we refer to [4] for a comprehensive introduction to such techniques in the context of shape optimization, and to Appendix A for a sketch of proof.

Theorem 2. *Assume that $\Omega \in \mathcal{O}_{ad}$ and that the kinematic viscosity ν is large enough. Then,*

(i) The energy dissipation $E(\Omega)$ given by (2.4) is shape differentiable and its derivative reads:

$$\forall \boldsymbol{\theta} \in \Theta_{ad},\ \ E'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \left(-2\nu e(\mathbf{u}) : e(\mathbf{u}) + 2\nu e(\mathbf{u}) : e(\mathbf{v}_e)\right) \boldsymbol{\theta} \cdot \mathbf{n}\, ds, \tag{2.9}$$

where $(\mathbf{v}_e, q_e)$ is an adjoint state, defined as the solution of the linear PDE

$$\begin{cases} -\nu\Delta \mathbf{v}_e + (\nabla \mathbf{u})^T \mathbf{v}_e - (\nabla \mathbf{v}_e)\mathbf{u} + \nabla q_e = -2\nu\Delta \mathbf{u} & \text{in } \Omega, \\ \operatorname{div}(\mathbf{v}_e) = 0 & \text{in } \Omega, \\ \mathbf{v}_e = 0 & \text{on } \Gamma \cup \Gamma_{in}, \\ \sigma(\mathbf{v}_e, q_e)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_e = 4\nu e(\mathbf{u})\mathbf{n} & \text{on } \Gamma_{out}. \end{cases} \tag{2.10}$$

(ii) the least-square functional $D(\Omega)$ defined by (2.5) is shape differentiable and its derivative reads:

$$\forall \boldsymbol{\theta} \in \Theta_{ad},\ \ D'(\Omega)(\theta) = \int_\Gamma 2\nu e(\mathbf{u}) : e(\mathbf{v}_d)\, \boldsymbol{\theta} \cdot \mathbf{n}\, ds, \tag{2.11}$$

where the adjoint system for $(\mathbf{v}_d, q_d)$ reads

$$\begin{cases} -\nu\Delta \mathbf{v}_d + (\nabla \mathbf{u})^T \mathbf{v}_d - (\nabla \mathbf{v}_d)\mathbf{u} + \nabla q_d = 0 & \text{in } \Omega, \\ \operatorname{div}(\mathbf{v}_d) = 0 & \text{in } \Omega, \\ \mathbf{v}_d = 0 & \text{on } \Gamma \cup \Gamma_{in}, \\ \sigma(\mathbf{v}_d, q_d)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_d = \mathbf{u} - \mathbf{u}_{ref} & \text{on } \Gamma_{out}. \end{cases} \tag{2.12}$$

Remark 3. Variational arguments ensure that the adjoint systems (2.10) and (2.12) are well-posed in suitable functional spaces when the parameter ν is assumed to be large enough (see e.g. [39]).

Like those of the functions $\mathrm{Vol}(\Omega)$, $\mathrm{Per}(\Omega)$, $E(\Omega)$ and $D(\Omega)$ involved in Theorems 1 and 2, the shape derivative of a fairly general class of shape functionals $F(\Omega)$ has the generic form:

$$F'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \phi\, \boldsymbol{\theta} \cdot \mathbf{n}\, ds =: (\phi, \boldsymbol{\theta}\cdot\mathbf{n})_{L^2(\Gamma)}, \tag{2.13}$$

where the scalar function $\phi : \Gamma \to \mathbb{R}$ is the 'shape gradient' of F with respect to the $L^2(\Gamma)$ inner product: it is the *structure theorem* for shape derivatives; see [37], §5.9. In particular, $F'(\Omega)(\boldsymbol{\theta})$ depends only on the values of the normal component $\boldsymbol{\theta}\cdot\mathbf{n}$ on the free boundary Γ; this reflects the intuitive fact that tangential deformations of Ω leave the values of $F(\Omega)$ unchanged at first order.

For further reference, the structure (2.13) makes it easy to calculate descent directions for $F(\Omega)$. Indeed, if $\boldsymbol{\theta}$ coincides with $-\phi\boldsymbol{n}$ on Γ, it readily follows from (2.8) that, for $t > 0$ small enough:

$$F(\Omega_{t\boldsymbol{\theta}}) = F(\Omega) - t\int_\Gamma \phi^2\, ds + o(t) < F(\Omega). \tag{2.14}$$

3. Numerical methods

In this section, we describe in more details the numerical methods involved in the resolution of the shape optimization problem (2.3).

3.1. Description of the numerical setting and outline of the algorithm.

Each shape Ω is represented by means of a simplicial mesh $\mathcal{T}$, composed of K (closed) simplices $T_1, ..., T_K$ (i.e. triangles in $2d$, tetrahedra in $3d$) , and I vertices $\mathbf{x}_1, ..., \mathbf{x}_I$. The mesh $\mathcal{T}$ is computational in the sense of Finite Elements, that is:

- The T_k form a cover of $\overline{\Omega}$, i.e. $\overline{\Omega} = \bigcup_{k=1}^K T_k$,
- The T_k do not overlap, i.e. the intersection between the interiors of T_k and $T_{k'}$ is empty whenever $k \neq k'$,
- The mesh $\mathcal{T}$ is *conforming*; for instance, in two dimensions, the intersection between any two triangles T_k and $T_{k'}$, $k \neq k'$, is either empty, or it is a vertex, or an edge of $\mathcal{T}$.

In the following, we shall often consider sequences of shapes Ω^n and meshes $\mathcal{T}^n$, and we denote with a ^{n} superscript all the entities (vertices $\mathbf{x}_i^n$, simplices T_k^n, numbers of vertices I^n and simplices K^n) of $\mathcal{T}^n$.

So as to evidence the needed numerical methods, we now give a deliberately hazy sketch of the main stages of the intended optimization process for (2.3); a practical version is given in Section 3.7.

- **Initialization:** The initial domain Ω^0 is equipped with a mesh $\mathcal{T}^0$.
- **For $n = 0$, ... until convergence:**
  (1) Calculate the solution $(\mathbf{u}, p)$ of the Navier-Stokes equation (2.1), and the adjoint state $(\mathbf{v}, q)$, solution of (2.10) or (2.12) on Ω^n, using the mesh $\mathcal{T}^n$.
  (2) Calculate the shape derivatives of $J(\Omega)$ and $G(\Omega)$ (see Theorems 1 and 2) and infer a descent direction $\boldsymbol{\theta}^n$ for the optimization problem (2.3).
  (3) Choose a sufficiently small time step τ^n and advect the shape Ω^n into the new shape $\Omega^{n+1} := (\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n)$; a mesh $\mathcal{T}^{n+1}$ of Ω^{n+1} is obtained.

This program raises a number of issues:

- The numerical resolution of the systems (2.1), (2.10) and (2.12) is by no means trivial; Section 3.2 below is devoted to this issue.
- The calculation of a descent direction for $J(\Omega)$ which allows to satisfy the constraint $G(\Omega)$ demands the use of an adapted optimization algorithm, which is described in Section 3.3.
- The deformation of the mesh $\mathcal{T}^n$ of Ω^n into a computational mesh $\mathcal{T}^{n+1}$ of Ω^{n+1} is a difficult task. We describe in Section 3.4 the stakes of mesh deformation, and in Section 3.5 a strategy for calculating a nice shape gradient which eases this purpose.

3.2. **Numerical resolution of the Navier-Stokes equations.**

The numerical resolution of the Navier-Stokes system (2.1) with the Finite Element method is faced with two relatively independent difficulties. The first one is related to the treatment of the nonlinear convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$; the second one is quite common in the resolution of saddle-point problems: it is about the choice of adequate Finite Element spaces for the discretization of the velocity $\mathbf{u}$ and pressure p. Notice that the adjoint systems (2.10) and (2.12) being linear, their resolution is not concerned by the first issue, but it is by the second one. We only discuss the case of the nonlinear Navier-Stokes system (2.1) - (2.2) in this section, which is on all points more difficult.

3.2.1. *Dealing with the nonlinear convective term using Newton's method.*

We rely on a fairly standard Newton method for nonlinear problems. Writing (3.3) in the abstract form

$$A(\mathbf{u}, p) = 0, \tag{3.1}$$

Newton's method achieves the solution as the limit of the sequence $(\mathbf{u}^k, p^k)$, where each update $(\delta\mathbf{u}^k, \delta p^k)$ between the steps k and $(k+1)$ is obtained by solving the linearized version of (3.1) at $(\mathbf{u}^k, p^k)$:

$$d_{(\mathbf{u}^k, p^k)} A(\delta\mathbf{u}^k, \delta p^k) = -A(\mathbf{u}^k, p^k), \tag{3.2}$$

where $d_{(\mathbf{u}^k, p^k)} A$ is the tangent operator to the mapping $(\mathbf{u}, p) \mapsto A(\mathbf{u}, p)$ at $(\mathbf{u}^k, p^k)$. In the particular case of interest for us, the iterative procedure given by (3.2) reads as follows

(1) The pair $(\mathbf{u}^0, p^0)$ is the solution to the Stokes system

$$\begin{cases} -\nu\Delta\mathbf{u}^0 + \nabla p^0 = 0 & \text{in } \Omega, \\ \mathrm{div}(\mathbf{u}^0) = 0 & \text{in } \Omega, \\ \mathbf{u}^0 = \mathbf{u}_{\mathrm{in}} & \text{on } \Gamma_{\mathrm{in}}, \\ \mathbf{u}^0 = 0 & \text{on } \Gamma, \\ \sigma(\mathbf{u}^0, p^0)\mathbf{n} = 0 & \text{on } \Gamma_{\mathrm{out}}. \end{cases} \tag{3.3}$$

(2) For $k = 1, ...$, until convergence, $(\mathbf{u}^{k+1}, p^{k+1})$ is obtained by

$$(\mathbf{u}^{k+1}, p^{k+1}) = (\mathbf{u}^k, p^k) + (\delta\mathbf{u}^k, \delta p^k),$$

where $(\delta\mathbf{u}^k, \delta p^k)$ is the solution to the system (viz. (3.2))

$$\text{(3.4)}\quad \begin{cases} -\nu\Delta(\delta\mathbf{u}^k) + (\mathbf{u}^k\cdot\nabla)(\delta\mathbf{u}^k) + ((\delta\mathbf{u}^k)\cdot\nabla)\mathbf{u}^k \ + \nabla(\delta p^k) = \nu\Delta\mathbf{u}^k - (\mathbf{u}^k\cdot\nabla)\mathbf{u}^k - \nabla p^k & \text{in } \Omega, \\ \operatorname{div}(\delta\mathbf{u}^k) = 0 & \text{in } \Omega, \\ \delta\mathbf{u}^k = 0 & \text{on } \Gamma\cup\Gamma_{\text{in}}, \\ \sigma(\delta\mathbf{u}^k, \delta p^k)\mathbf{n} = 0 & \text{on } \Gamma_{\text{out}}. \end{cases}$$

(3) The procedure ends when the following convergence criterion is fulfilled:

$$\text{(3.5)}\qquad \frac{||u_1^k||_{L^2(\Omega)}}{||u_1^0||_{L^2(\Omega)}} + \frac{||u_2^k||_{L^2(\Omega)}}{||u_2^0||_{L^2(\Omega)}} + ... + \frac{||u_d^k||_{L^2(\Omega)}}{||u_d^0||_{L^2(\Omega)}} + \frac{||p^k||_{L^2(\Omega)}}{||p^0||_{L^2(\Omega)}} < \varepsilon_{\text{Newton}}$$

for a fixed, user-defined tolerance $\varepsilon_{\text{Newton}}$.

Let us mention that many other methods are available for the numerical resolution of (2.1), such as the Oseen iteration method, the Least-Square gradient method, the Peaceman-Racheford method (an increment of the Least-Square gradient method), with different assets and drawbacks which we do not discuss here; see [33] about this purpose.

On a different note, Newton-like algorithms are well-known to experience difficulties as far as convergence is concerned; in the context of the numerical resolution of the Navier-Stokes system (2.1), this is likely to happen in the case of moderate-to-high Reynolds numbers, where the initial guess $(\mathbf{u}^0, p^0)$ may be 'too far' from the sought solution $(\mathbf{u}, p)$ of (2.1). In such a case, one may resort to mixed strategies (e.g. starting with the Oseen iteration method, then branching with the Newton method), or continuation methods (which advocate to increase steadily the Reynolds number) to improve and make robuster the convergence process. As we have already mentioned, the model examples considered in this article (see Section 4) arise in the regime of low Reynolds number, and we did not run into the need for such elaborated strategies.

3.2.2. *Choice of the Finite Element discretization.*

When it comes to the numerical resolution of linear saddle point problems of the form (3.3) or (3.4), one should pay attention to the choice of the Finite Element spaces used for the discretization of the unknown velocity $\mathbf{u}$ and pressure p. In our case, (3.3) and (3.4) are solved with the Finite Element method in mixed velocity-pressure formulation, using $\mathbb{P}_2$ Lagrange elements for the velocity $\mathbf{u}$ and $\mathbb{P}_1$ Lagrange elements for the pressure p. This choice as regards Finite Element spaces is one among those ensuring that the so-called *Brezzi inequality* holds, and thereby that the discrete linear systems corresponding to (3.3) and (3.4) are invertible. Details about numerical methods for the resolution of saddle point problems can be found in [11] or [28].

3.3. **The augmented Lagrangian algorithm for equality-constrained problems.**

In order to drive the numerical resolution of (2.3), we rely on the augmented Lagrangian method, a basic sketch of which is now provided; we refer to [50] §17.4 for detailed explanations.

The augmented Lagrangian algorithm transforms the constrained optimization problem (2.3) into the series of unconstrained problems (hereafter indexed by the superscript ^{n}):

$$\text{(3.6)}\qquad \inf_{\Omega\in\mathcal{O}_{ad}} \mathcal{L}(\Omega, \ell^n, b^n), \text{ where } \mathcal{L}(\Omega, \ell, b) = J(\Omega) - \ell G(\Omega) + \frac{b}{2}G(\Omega)^2.$$

In the definition of the augmented Lagrangian $\mathcal{L}$, the parameter b is a (positive) penalty factor for the violation of the constraint $G(\Omega) = 0$, and ℓ is an estimate of the Lagrange multiplier associated to this constraint in (2.3).

The augmented Lagrangian algorithm intertwines the search for the minimizer Ω^n of $\Omega \mapsto \mathcal{L}(\Omega, \ell^n, b^n)$ for fixed values of ℓ^n and b^n, and the update of these coefficients according to the rule:

$$\text{(3.7)}\qquad \ell^{n+1} = \ell^n - b^n G(\Omega^n), \text{ and } b^{n+1} = \begin{cases} \alpha b^n & \text{if } n < n_0, \\ b^n & \text{otherwise,} \end{cases}$$

where $\alpha > 1$ denotes a fixed parameter.

We again refer to [50] for an insight about this procedure; let us simply mention that ℓ^n is an increasingly accurate approximation of the Lagrange multiplier for the constraint $G(\Omega) = 0$ featured in (2.3). Notice also

that the penalty coefficient b^n is multiplied by a user-defined constant $\alpha > 1$ during the first n_0 iterations of the algorithm, and that it is kept fixed afterwards. In particular, the augmented Lagrangian strategy does not require b^n to tend to infinity so that the constraint $G(\Omega) = 0$ be fulfilled; this guarantees a better conditioning of (3.6) with respect to the naive quadratic penalty method (featuring only the first and last terms in the definition of $\mathcal{L}$ in (3.6)).

In our context, where the computational burden of minimizing $\Omega \mapsto \mathcal{L}(\Omega, \ell, b)$ is significant, we rely on the following practical implementation of these ideas which limits the number of iterations of the method.

- **Initialization:** Start from an initial shape Ω^0 and coefficients ℓ^0 and b^0.
- **For** $n = 0, ...$ **until convergence**
  - choose a descent direction $\boldsymbol{\theta}^n$ for $\Omega \mapsto \mathcal{L}(\Omega, \ell^n, b^n)$,
  - take τ^n small enough so that $\mathcal{L}((\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n), \ell^n, b^n) < \mathcal{L}(\Omega^n, \ell^n, b^n)$, and set $\Omega^{n+1} = (\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n)$.
  - update the coefficients ℓ^n and b^n of the augmented Lagrangian $\mathcal{L}$ according to (3.7).

3.4. **Mesh-related issues.**

Assume for one moment that a descent direction $\boldsymbol{\theta}^n$ for (2.3) and a descent step τ^n have been found at the n^{th} iteration of the procedure described in Section 3.1; we are faced with the realization of the operation $\Omega^n \mapsto \Omega^{n+1} = (\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n)$. If $\mathcal{T}^n$ is the mesh of Ω^n, the natural way to carry it out reads:

$$\mathbf{x}_i^n \mapsto \mathbf{x}_i^{n+1} := \mathbf{x}_i^n + \tau^n \boldsymbol{\theta}^n(\mathbf{x}_i^n), \;\; i = 1, ..., I^n, \tag{3.8}$$

while the connectivities of the mesh are unchanged, i.e. the considered mesh $\mathcal{T}^{n+1}$ of Ω^{n+1} is made of the same simplices than $\mathcal{T}^n$, but their vertices are relocated according to $\boldsymbol{\theta}^n$.

Unfortunately, this simple procedure is likely to give rise to very stretched elements within a few iterations. This is a pity since the accuracy of the resolution of PDE with the Finite Element method greatly depends on the *quality* of the elements in the mesh, i.e. on their being close to equilateral [20]. Hence, the numerical resolution of (2.1) may become very inaccurate (not to say impossible), causing the optimization process to stop prematurely. To circumvent this drawback, we rely on two ingredients:

- The emergence of stretched elements in $\mathcal{T}^n$ may be postponed: in the transformation of Ω^n into Ω^{n+1} (practically, that of $\mathcal{T}^n$ into $\mathcal{T}^{n+1}$ via (3.8)), only the values of $\boldsymbol{\theta}^n$ on the boundary Γ^n determine the new domain Ω^{n+1}; in the numerical framework, the values of $\boldsymbol{\theta}^n$ inside Ω^n are only used to relocate the internal vertices of $\mathcal{T}^n$. In particular, these values may be freely chosen in a way that makes $\mathcal{T}^{n+1}$ of good quality insofar as possible, as we describe in the next Section 3.5.
- When the quality of the mesh becomes poor, i.e. in our context when

  $$\min_{k=1,...,K^n} |T_k^n| < \varepsilon_{\text{mesh}},$$

  where $\varepsilon_{\text{mesh}}$ is a user-defined parameter (see the discussion in [30] §18.2 about other possible quality measures of a mesh, which could be easily implemented in FreeFem++), a local remeshing of $\mathcal{T}^n$ is carried out: in a nutshell,
  - 'Too long' edges are split,
  - The endpoints of 'too short' edges are merged,
  - The connectivities of ill-shaped triangles (e.g. nearly flat triangles) are swapped,
  - Vertices are slightly relocated,

  as long as the overall quality of the mesh is improved; see [30] Chap. 18 for more details.

3.5. **Extension-regularization of the shape gradient.**

As we have seen, our optimization procedure amounts to a series of minimizations of functionals of the form (3.6), which we generically denote by $F(\Omega)$ in the present section. We have seen that a natural candidate for a descent direction is

$$\boldsymbol{\theta} = -\phi \, \mathbf{n}, \tag{3.9}$$

where the scalar function $\phi : \Gamma \to \mathbb{R}$ is the $L^2(\Gamma)$-shape gradient of $F(\Omega)$, supplied by the shape derivative of $F(\Omega)$ via (2.13).

Unfortunately, this choice is generally ill-suited for at least two reasons:

(i) Strictly speaking, (3.9) only makes sense on the boundary Γ of the actual shape Ω, while the numerical setting requires the velocity field $\boldsymbol{\theta}$ to be defined on Ω as a whole, see (3.8).

(ii) The $L^2(\Gamma)$ shape gradient ϕ of $F(\Omega)$ may be very irregular, especially in the areas surrounding Γ_{out} because of the change in boundary conditions occurring there. This may cause numerical artefacts when it comes to the mesh procedure (3.8); see for instance [46] §6.2.4. It is therefore often desirable to smooth the velocity field $\boldsymbol{\theta}$ on Γ before performing (3.8).

The popular *extension-regularization* procedure overcomes both difficulties in such a way that $\boldsymbol{\theta}$ stays a descent direction for $F(\Omega)$; see e.g. [16, 22, 26]. The basic idea consists in identifying a shape gradient for $F(\Omega)$ from its shape derivative $F'(\Omega)(\boldsymbol{\theta})$ (see (2.13)) by means of a different inner product $(\cdot,\cdot)_V$ than $(\cdot,\cdot)_{L^2(\Gamma)}$, acting on a (Hilbert) space V of more regular vector fields, defined on Ω as a whole. More precisely, one searches for $\boldsymbol{\theta} \in V$ such that

$$\forall \boldsymbol{\psi} \in V,\ \ (\boldsymbol{\theta}, \boldsymbol{\psi})_V = J'(\Omega)(\boldsymbol{\psi}) = \int_\Gamma \phi\, \boldsymbol{\psi} \cdot \mathbf{n}\, ds. \tag{3.10}$$

Doing so ensures that:

$$J'(\Omega)(-\boldsymbol{\theta}) = -(\boldsymbol{\theta}, \boldsymbol{\theta})_V < 0,$$

which together with (2.14) guarantees that $\boldsymbol{\theta}$ is also a descent direction for $F(\Omega)$.

To be quite precise, in our context, we rely on the space

$$V = \{\mathbf{v} \in H^1(\Omega)^d, \mathbf{v}|_{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}} = 0,\ \nabla_\Gamma \mathbf{v} \in L^2(\Gamma)^d\},$$

where $\nabla_\Gamma f := \nabla f - (\nabla f \cdot \mathbf{n})\mathbf{n}$ is the tangential gradient of a (smooth) function f; V is equipped with the inner product

$$(\boldsymbol{\theta}, \boldsymbol{\psi})_V = \gamma \int_\Omega Ae(\boldsymbol{\theta}) : e(\boldsymbol{\psi})\, dx + (1-\gamma) \int_\Gamma \nabla_\Gamma \boldsymbol{\theta} \cdot \nabla_\Gamma \boldsymbol{\psi}\, ds. \tag{3.11}$$

This definition features two contributions, balanced by the parameter $\gamma \in [0, 1]$:

- The first term in (3.11) is inspired from the linearized elasticity equations. Here, A is the Hooke's law, acting on symmetric matrices e with size $d \times d$,

  $$Ae = 2\mu e + \lambda \text{tr}(e),$$

  where λ and μ are the Lamé coefficients of the fictitious elastic material. This choice - which is widespread in meshing [9, 25] to help in keeping a mesh with fine quality - is motivated by the intuition that elastic displacements tend to induce little compression (i.e. local change in the volume).
- The second term in (3.11) corresponds to the Laplace-Beltrami operator on Γ; its role is to impose that the considered vector fields enjoy more regularity on Γ than mere $L^2(\Gamma)$ regularity.

With these definitions at hand, the desired 'regularized' shape gradient $\boldsymbol{\theta}$ is calculated by solving (3.10) with a standard Finite Element method on a mesh of Ω.

Remark 4. In our implementation, the Lamé parameters λ, μ of the elastic material used for the extension-regularization procedure are homogeneous over Ω. Notice that the above strategy could be easily improved by considering *inhomogeneous* elasticity coefficients λ, μ, for instance coefficients characterized by a larger Young's modulus (which measures the resistance to traction and compression efforts) in regions where the mesh of Ω has stretched elements, so to penalize the relative compression rate they undergo.

Remark 5. A perhaps more natural idea consists in choosing $V = \left\{\mathbf{v} \in H^1(\Omega)^d,\ \mathbf{v}|_{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}} = 0\right\}$, with associated inner product:

$$(\boldsymbol{\theta}, \boldsymbol{\psi})_V = \gamma \int_\Omega \nabla \boldsymbol{\theta} : \nabla \boldsymbol{\psi}\, dx + \int_\Omega \boldsymbol{\theta} \cdot \boldsymbol{\psi}\, dx,$$

where $\gamma > 0$ is a 'small' parameter. In this context, (3.10) amounts to solving the regularizing, elliptic system:

$$\begin{cases} -\gamma \Delta \boldsymbol{\theta} + \boldsymbol{\theta} = 0 & \text{in } \Omega, \\ \boldsymbol{\theta} = 0 & \text{on } \Gamma_{\text{in}} \cup \Gamma_{\text{out}}, \\ \gamma \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{n}} = -\phi \mathbf{n} & \text{on } \Gamma. \end{cases} \tag{3.12}$$

However easy to implement, this choice is less efficient than (3.11) insofar as it does not show the same efficiency in preventing the emergence of stretched elements; see the example in Section 4.1 about this point.

3.6. **Calculation of the curvature.**

Most of the numerical methods involved in the resolution of the shape optimization problem (2.3) imply the calculations of the normal vector $\mathbf{n}$ and curvature κ of the boundary $\partial\Omega$ of a shape Ω (see for instance Theorem 1). In practice, these quantities are evaluated from the discrete geometry of a mesh $\mathcal{T}$ of Ω, which is not a completely straightforward task. In this section, following [30], we describe a simple, yet robust method to achieve this goal in the case of two space dimensions: $d = 2$. Similar approximations hold in the general case, which involve more tedious notations.

Let $\mathbf{x}_i$ be a vertex of $\mathcal{T}$ lying on $\partial\Omega$, and let $\mathbf{x}_{i-1}$ (resp. $\mathbf{x}_{i+1}$) be the vertex on $\partial\Omega$ located immediately before (resp. after) $\mathbf{x}_i$ when $\partial\Omega$ is oriented counterclockwise; see Figure 3.

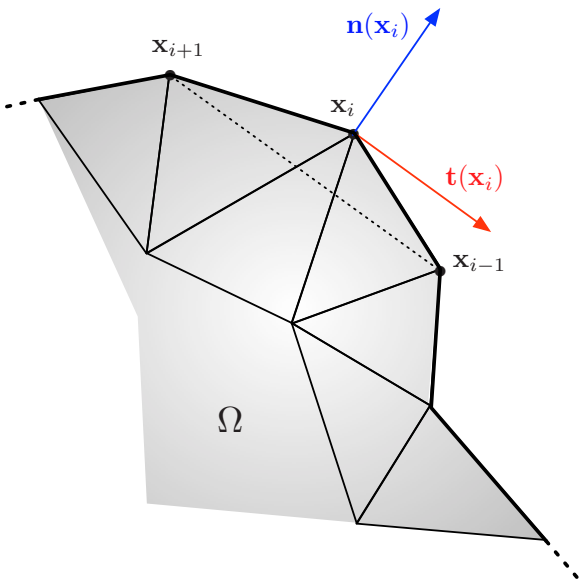

FIGURE 3. *Calculation of the tangent and normal vectors to $\partial\Omega$ from the data of a triangular mesh.*

In this situation, the tangent vector $\mathbf{t}(\mathbf{x}_i)$ to $\partial\Omega$ at $\mathbf{x}_i$ is calculated as:

$$\mathbf{t}(\mathbf{x}_i) = \frac{\overrightarrow{\mathbf{x}_{i+1}\mathbf{x}_{i-1}}}{|\overrightarrow{\mathbf{x}_{i+1}\mathbf{x}_{i-1}}|},$$

and the unit normal vector $\mathbf{n}(\mathbf{x}_i)$ to $\partial\Omega$ at $\mathbf{x}_i$, pointing outward Ω is estimated as the rotate of $\mathbf{t}(\mathbf{x}_i)$:

$$\mathbf{n}(\mathbf{x}_i) = \begin{pmatrix} -t_2(\mathbf{x}_i) \\ t_1(\mathbf{x}_i) \end{pmatrix}.$$

Thence, the radius of curvature at $\mathbf{x}_i$ is approximated as:

$$r(\mathbf{x}_i) = \frac{1}{4}\left(\frac{\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}} \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}}{-\mathbf{n}(\mathbf{x}_i) \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}} + \frac{\overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}} \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}}{-\mathbf{n}(\mathbf{x}_i) \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}} \right), \tag{3.13}$$

and the curvature $\kappa(\mathrm{x}_i)$ at $\mathbf{x}_i$ is obtained as $\kappa(\mathrm{x}_i) = \frac{1}{r(\mathbf{x}_i)}$ if none of the denominators featured in (3.13) equals 0 (and it is set to 0 otherwise).

3.7. **Algorithmic description of the implemented method.**

We are now ready to describe the shape optimization algorithm arising from the previous considerations.

(1) **Initialization.**
The initial shape $\Omega^0 \in \mathcal{O}_{ad}$ is equipped with a triangular mesh $\mathcal{T}^0$. Select initial values for the coefficients ℓ^0, $b^0 > 0$ of the augmented Lagrangian algorithm.

(2) **Main loop: for** $n = 0, ...$

(i) Calculate the solution $(\mathbf{u}^n, p^n)$ of the Navier-Stokes system (2.1) on the mesh $\mathcal{T}^n$ of Ω^n by using the material in Section 3.2.

(ii) Calculate the solution $(\mathbf{v}^n, q^n)$ of the adjoint system (2.10) or (2.12) on Ω^n.

(iii) Calculate the $L^2(\Gamma^n)$ shape gradient ϕ^n of $\Omega \mapsto \mathcal{L}(\Omega, \ell^n, b^n)$ by using Theorem 2.

(iv) Infer a descent direction $\boldsymbol{\theta}^n$ for $\Omega \mapsto \mathcal{L}(\Omega, \ell^n, b^n)$ by solving (3.10) on the mesh $\mathcal{T}^n$.

(v) Find a descent step τ^n such that

$$\mathcal{L}((\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n), \ell^n, b^n) < \mathcal{L}(\Omega^n, \ell^n, b^n) \text{ (possibly up to a small tolerance) .} \tag{3.14}$$

(vi) Move the vertices of $\mathcal{T}^n$ according to τ^n and $\boldsymbol{\theta}^n$:

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \tau^n \boldsymbol{\theta}^n(\mathbf{x}_i^n) \tag{3.15}$$

- If the resulting mesh is invalid, go back to step (v), and use a smaller value for τ^n,
- Else, the positions (3.15) define the vertices of the new mesh $\mathcal{T}^{n+1}$.

(vii) If the quality of $\mathcal{T}^{n+1}$ is too low, use a local remeshing as described in Section 3.4.

(viii) Update the augmented Lagrangian parameters according to (3.7).

(3) **Ending criterion.** Stop if

$$||\boldsymbol{\theta}^n||_{L^2(\Gamma^n)} < \varepsilon_{\text{stop}}$$

Return Ω^n.

4. Numerical illustrations

In this section, we present five two-dimensional applications of the numerical algorithm presented in Section 3.7. The geometric configurations associated to these examples are represented in Figure 4, and the parameters used in the different test cases (initial parameters of the augmented Lagrangian algorithm, target volume, etc.) are reported in Table 1.

Description	Parameter	Case 1	Case 2	Case 3	Case 4	Case 5
Imposed volume over initial volume	$\frac{V_T}{V_0}$	1	1		1	
Imposed perimeter over initial perimeter	$\frac{P_T}{P_0}$					0.97
Regularization weight	γ	10^{-2}	10^{-2}	1	1	1
Initial Lagrange multiplier	ℓ^0	0	0	0	15	0
Lagrange multiplier update coefficient	α	1.05	1.05	1.05	1.05	1.05
Initial gradient step size	τ^0	10^{-2}	10^{-2}	10^{-2}	$3 * 10^{-3}$	10^{-2}
Stopping criterion	ε_{stop}	10^{-2}	$5 * 10^{-3}$		$2 * 10^{-2}$	10^{-2}
Reynolds number	Re	200	200	200	200	200

Table 1. Parameters used for the numerical examples of Section 4.

4.1. **Minimization of the dissipated energy in a bend.**

Our first benchmark example is concerned with the optimization of the shape of a pipe with orthogonal inlet and outlet, as depicted in Figure 4 (a); see for instance [21, 13]. In a nutshell, this test case answers the question:

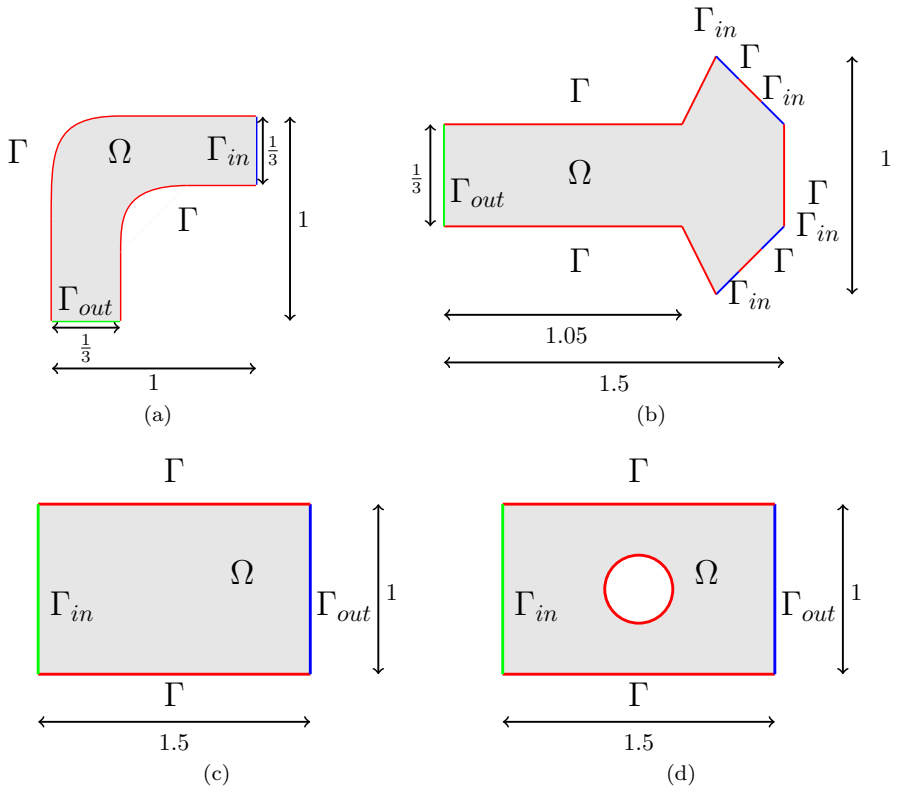

FIGURE 4. *Settings of the five test cases discussed in Section 4; (a) the bend, discussed in Section 4.1, (b) the ramified structure of Sections 4.2 and 4.3, (c) the straight pipe with one inlet, one outlet where a least-square criterion is considered, as studied in Section 4.4, (d) the dissipated energy minimization example of Section 4.5.*

"How to build a pipe with fixed volume that spends the least amount of energy to convey a fluid from Γ_{in} to Γ_{out}?"

The inlet flow is given by the parabolic profile

$$\mathbf{u}_{\text{in}}(x_1, x_2) = ((1 - x_2)(\frac{2}{3} - x_2), 0).$$

Starting from the initial shape Ω^0 represented in Figure 6 (top-left), we minimize the work of viscous forces, i.e. $J(\Omega) = E(\Omega)$, as defined by (2.4), under the volume constraint $\text{Vol}(\Omega) = \text{Vol}(\Omega^0)$, i.e. the target volume is that of Ω^0.

The results are displayed on Figure 6, and the associated convergence histories are included in Figure 5. The dissipated viscous energy decreases by roughly 25% during the process, and as expected, the optimized design tends to approach a straight pipe.

Let us emphasize the role of the regularizing parameter γ featured in the definition of the extension-regularization inner product (3.11). In this example, the $L^2(\Gamma)$ shape gradient of $E(\Omega)$ is not smooth in the vicinity of Γ_{out}. Therefore, if no regularization of this gradient is applied, mesh intersections appear within a few iterations in this region.

Eventually, let us mention that this test case is fairly insensitive to the computation parameters ℓ^0 and τ, which makes it the easiest of all five to run.

4.2. **Minimization of the dissipated energy in a ramified structure with volume constraint.**

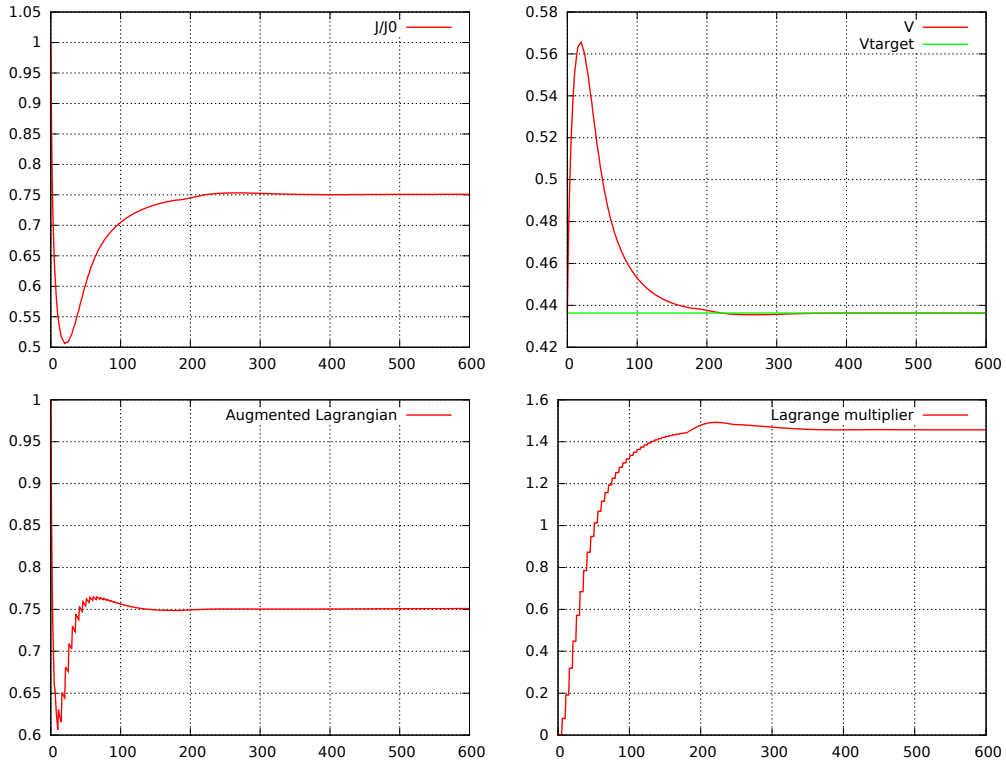

FIGURE 5. *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega,\ell,b)$ *and* ℓ^n *in the bend optimization example of Section 4.1*

Our second example is a simple model for the ramified structure of a human lung. It can be considered as an extension of the study in [23], where the same kind of problem is studied.

The situation is that of Figure 8 (top-right), where parabolic profiles are imposed on each component of Γ_{in}. More precisely, each such component is a line segment joining two points (x_1^A, x_2^A) and (x_1^B, x_2^B), which are distributed counterclockwise on Γ_{in}. The imposed inlet flow on each such line segment is defined by:

$$\mathbf{u}_{\mathrm{in}}(x_1,x_2) = s(1-s)\begin{pmatrix} -(x_2^B - x_2^A) \\ x_1^B - x_1^A \end{pmatrix}, \text{ where } s = \frac{x_1^A - x_1}{x_1^A - x_1^B} = \frac{x_2^A - x_2}{x_2^A - x_2^B}, \tag{4.1}$$

so that *ie* $\mathbf{u}\cdot\mathbf{n} \le 0$ on Γ_{in}.

We again aim at optimizing the energy dissipated owing to viscous effects, i.e. $J(\Omega) = E(\Omega)$, under the volume constraint $\mathrm{Vol}(\Omega) = V_T$, where $V_T = \mathrm{Vol}(\Omega_0)$.

The results are presented in Figure 7, and the associated convergence histories are those in Figure 8. Interestingly enough, ramifications appear in the course of the iterations and the optimized shape is much smoother than the initial one.

This example shows large mesh deformations, which justifies the importance of using a good extension-regularization procedure, such as that introduced in section 2.3. Notice that using the simpler procedure (3.12) in this example leads to mesh intersections after only a few iterations.

4.3. **Minimization of the dissipated energy in a ramified structure with perimeter constraint.**

This third example arises in the exact same physical context as that of Section 4.2 (again, see Figure 8 (top-right)). The only difference with the latter is that we now impose a constraint on the perimeter of shapes: $\mathrm{Per}(\Omega) = 0.97\,\mathrm{Per}(\Omega^0)$. The convergence histories of the computation are reported on Figure 9, and the shape at several intermediate stages is represented on Figure 10.

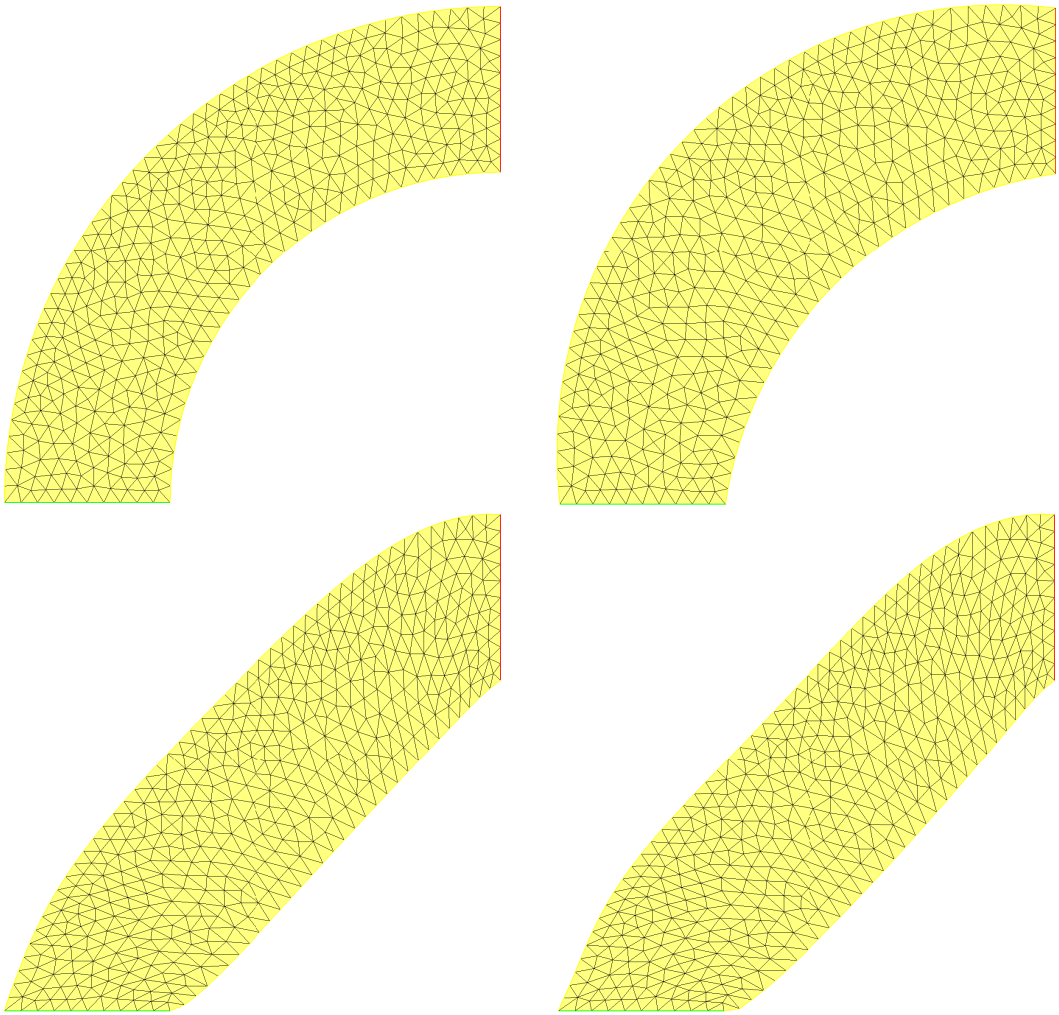

FIGURE 6. *Intermediate shapes Ω^n obtained in the bend optimization example of Section 4.1 at iterations (from left to right, top to bottom) $n = 0, 5, 100$ and 500.*

4.4. **Minimization of the discrepancy with a reference velocity profile.**

Our third example considers pipes Ω in the situation depicted on Figure 4 (bottom-left), where the parabolic profile

$$\mathbf{u}_{\text{in}}(x_1, x_2) = (x_2(1 - x_2), 0)$$

is imposed on the inlet Γ_{in}. Our aim is to optimize the shape of Ω with respect to the least-square criterion $J(\Omega) = D(\Omega)$ given by:

$$D(\Omega) = \int_{\Gamma_{\text{out}}} |\mathbf{u} - \mathbf{u}_{\text{ref}}|^2 \, ds, \tag{4.2}$$

where the reference profile $\mathbf{u}_{\text{ref}}$ is slightly off-centered (see Figure 12 (left)), namely:

$$\mathbf{u}_{\text{ref}}(x_1, x_2) = (2x_2^2(1 - x_2), 0). \tag{4.3}$$

This example is motivated by the following consideration: the initial shape Ω^0 is a straight cylinder, and it is well-known that the laminar flow satisfying the Navier-Stokes equations (2.1) is of Poiseuille type ([48], chapter 6) :

$$\mathbf{u}(x_1, x_2) = (x_2(1 - x_2), 0). \tag{4.4}$$

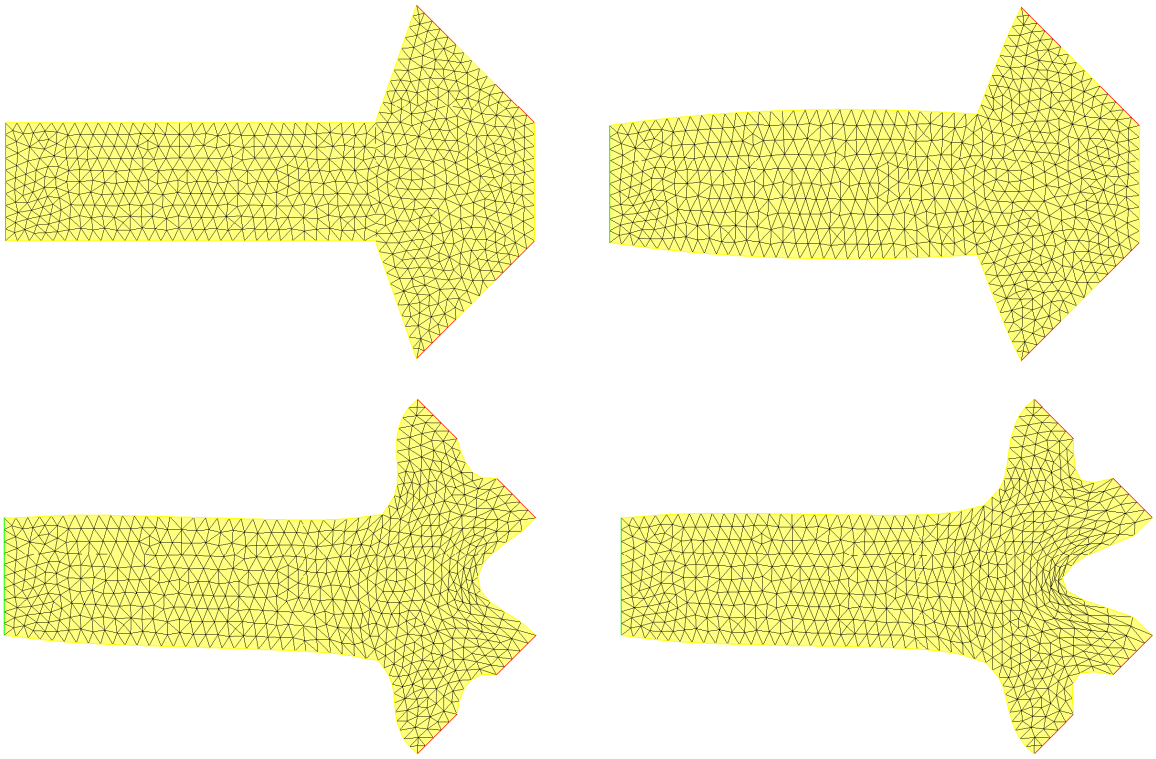

FIGURE 7. *From left to right, top to bottom, successive shapes* Ω^n *at iterations* $n = 0, 5, 240, 1000$ *in the dissipated energy minimization example in a ramified structure with volume constraint of Section 4.2.*

Hence, imposing that the velocity $\mathbf{u}$ in the pipe resemble (4.3) on Γ_{out} should shift the maximum of the horizontal component u_1 towards the top of the pipe.

No constraint is applied, so that the algorithm of Section 3.7 reduces to a simple gradient method in this case, i.e. $b = \ell_0 = 0$.

Several intermediates shapes are presented on Figure 13, and the corresponding one-dimensional profiles of u_1 and u_2 on Γ_{out} are reported on Figure 12. The least-square criterion $D(\Omega)$ decreases by roughly 63% in the course of the optimization. To achieve this decrease, the shape Ω develops a 'bump' on its bottom side, so that the flow is deviated towards the top. After iteration 200, a cusp appears at the bottom-right corner of Ω, which interrupts the algorithm. However, at this stage, $J(\Omega^n)$ is only a few percents lower than $J(\Omega^{200})$, so we decided to stop the optimization process at iteration 200.

4.5. **Energy dissipation around an obstacle.**

In our last example, depicted on Figure 4 (bottom-right), a solid obstacle is immersed in a cavity filled with a fluid, and the shapes Ω stand for the fluid domain, which is the complement of the obstacle in the cavity. Our aim is to minimize the dissipated energy in the cavity with respect to the shape of the obstacle, i.e. $J(\Omega) = E(\Omega)$.

This test-case is a simplified version of the three-dimensional optimization problem of the shape of a heat exchanger, where the thermal part is not taken into account. Many studies of this problem were performed [53],[24]. In the full problem, the obstacle stands for the section of a pipe carrying a warmer or colder fluid, and one aims at minimizing the dissipated energy under the constraint that minimum heat is exchanged between both fluids.

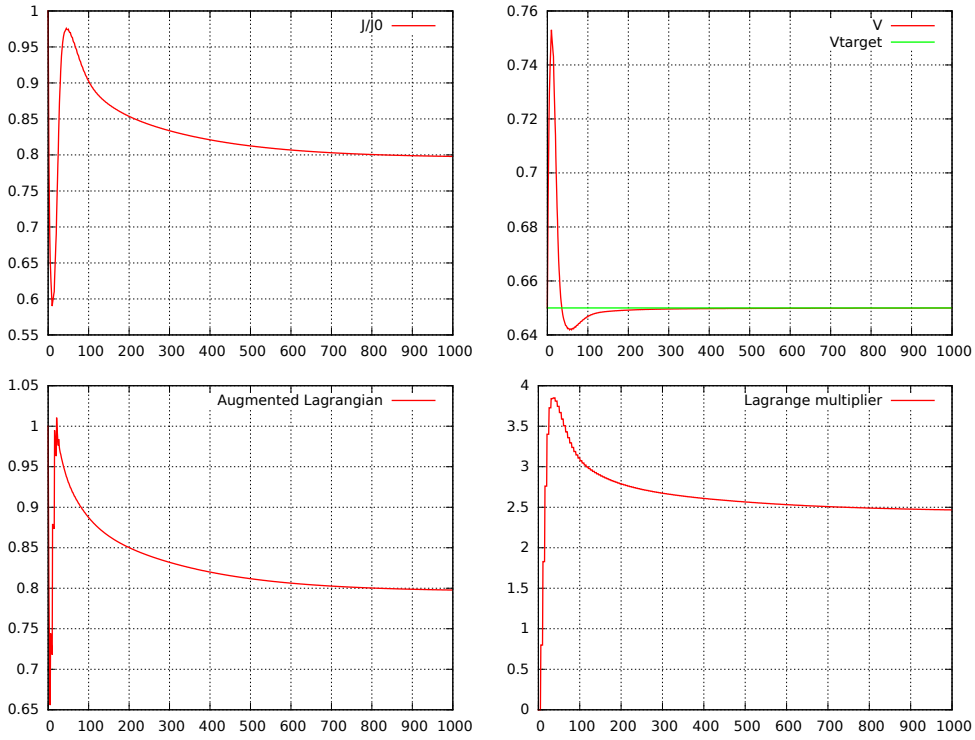

FIGURE 8. *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega,\ell,b)$ *and* ℓ^n *in the dissipated energy minimization example in a ramified structure of Section 4.2.*

In the model situation discussed here, we impose a horizontal flow on $\Gamma_{\rm in}$, namely

$$\mathbf{u} = \mathbf{u}_{\rm in}(x_1,x_2) = (1,0) \quad \text{on } \Gamma_{\rm in},$$

and no slip boundary conditions are prescribed on the boundary of the obstacle.

The optimized shape of the obstacle resembles a wing; see Figure 15 (bottom) The convergence histories are presented on Figure 14. Let us mention that, from the numerical point of view, this test-case is the hardest to run, since ℓ_0 and τ have to be chosen carefully. From a practical point of view, this choice relies on a few trials on very coarse meshes.

5. Conclusion and perspectives

In this article, we have presented a numerical framework for shape optimization in the context of fluid mechanics, consisting of well-established techniques which we have strived to present in an elementary and pedagogical way. The discussed numerical examples have been implemented in the `FreeFem++` environment; the source code and a short manual are available online at `https://github.com/flomnes/optiflow`, and can be easily handled and extended to deal with other geometric situations, or shape optimization problems.

Acknowledgements. Y. Privat was partially supported by the Project "Analysis and simulation of optimal shapes - application to lifesciences" of the Paris City Hall.

Appendix A. Sketch of the proof of Theorem 2

The differentiability of the solution $(\mathbf{u},p)$ to the Navier-Stokes system (2.1) with respect to the domain is a technical, albeit quite classical matter, and we admit the result, referring for instance to [37] for the rigorous definition of this notion, and to [39] or [23] for this precise calculation. The derivative $(\mathbf{u}',p')$ of

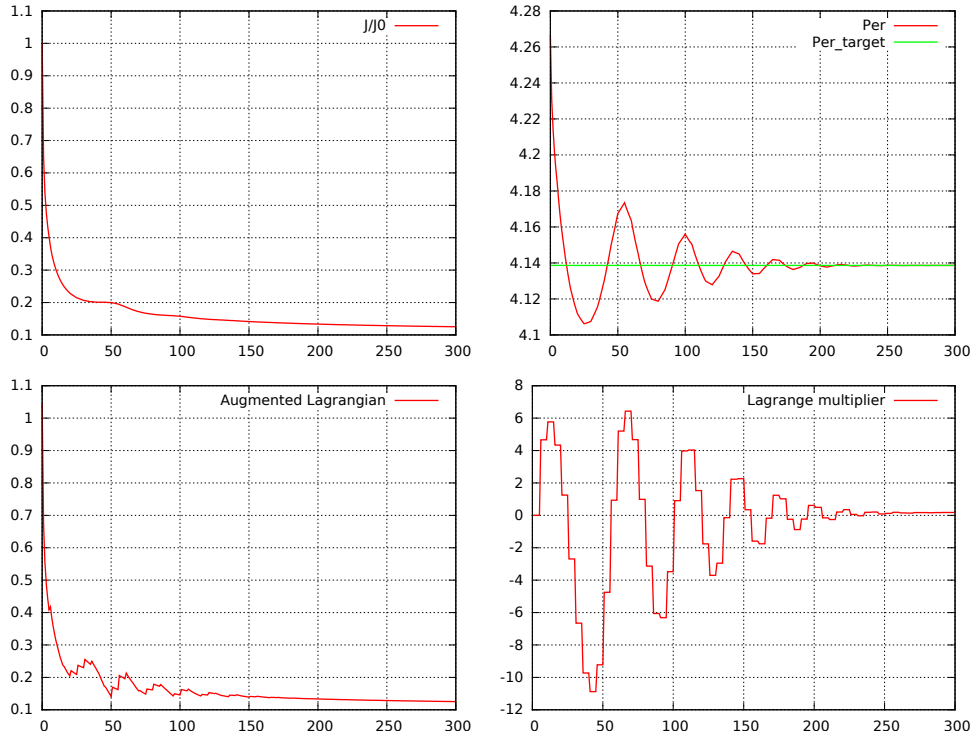

FIGURE 9. *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Per}(\Omega)$, $\mathscr{L}(\Omega,\ell,b)$ *and* ℓ^n *in dissipated energy minimization in a ramified structure example of Section 4.3*

$(\mathbf{u},p)$ with respect to the domain, in the direction $\boldsymbol{\theta}\in\Theta_{ad}$, is solution to the problem:

$$\text{(A.1)}\qquad \begin{cases} -\nu\Delta\mathbf{u}'+(\nabla\mathbf{u})\mathbf{u}'+(\nabla\mathbf{u}')\mathbf{u}+\nabla p'=0 & \text{in } \Omega,\\ \mathrm{div}(\mathbf{u}')=0 & \text{in } \Omega,\\ \sigma(\mathbf{u}',p')\mathbf{n}=0 & \text{on } \Gamma_{\mathrm{out}},\\ \mathbf{u}'=0 & \text{on } \Gamma_{\mathrm{in}},\\ \mathbf{u}'=-\left(\frac{\partial\mathbf{u}}{\partial n}\right)(\boldsymbol{\theta}.n) & \text{on } \Gamma. \end{cases}$$

Also, we only present the calculation of the shape derivative of the functional $D(\Omega)$ given by (2.5), the calculation being on any point easier in the case of $E(\Omega)$; see [23] if need be.

Using the chain rule from the definition (2.5) of $D(\Omega)$ yields:

$$\text{(A.2)}\qquad D'(\Omega)(\boldsymbol{\theta})=\int_{\Gamma_{\mathrm{out}}}\mathbf{u}'\cdot(\mathbf{u}-\mathbf{u}_{\mathrm{ref}})\,ds.$$

The main idea of the proof consists in using the adjoint state $(\mathbf{v}_d,q_d)$, solution to (2.12): performing several integrations by parts allows to eliminate the unknown derivatives $(\mathbf{u}',p')$ from the expression (A.2). More precisely, multiplying the first equation in (A.1) by $\mathbf{v}_d$ and integrating by parts yields

$$\begin{aligned} 0&=\int_\Omega(-\nu\Delta\mathbf{u}'+(\nabla\mathbf{u})\mathbf{u}'+(\nabla\mathbf{u}')\mathbf{u}+\nabla p')\cdot\mathbf{v}_d\,dx,\\ &=\int_\Omega(2\nu\,e(\mathbf{u}'):e(\mathbf{v}_d)-\mathrm{div}(\mathbf{v}_d)p'+(\nabla\mathbf{u})\mathbf{u}'\cdot\mathbf{v}_d+(\nabla\mathbf{u}')\mathbf{u}\cdot\mathbf{v}_d)\,dx-\int_{\partial\Omega}\sigma(\mathbf{u}',p')\mathbf{n}\cdot\mathbf{v}_d\,ds,\\ &=\int_\Omega(2\nu\,e(\mathbf{u}'):e(\mathbf{v}_d)+(\nabla\mathbf{u})\mathbf{u}'\cdot\mathbf{v}_d+(\nabla\mathbf{u}')\mathbf{u}\cdot\mathbf{v}_d)\,dx, \end{aligned}\qquad\text{(A.3)}$$

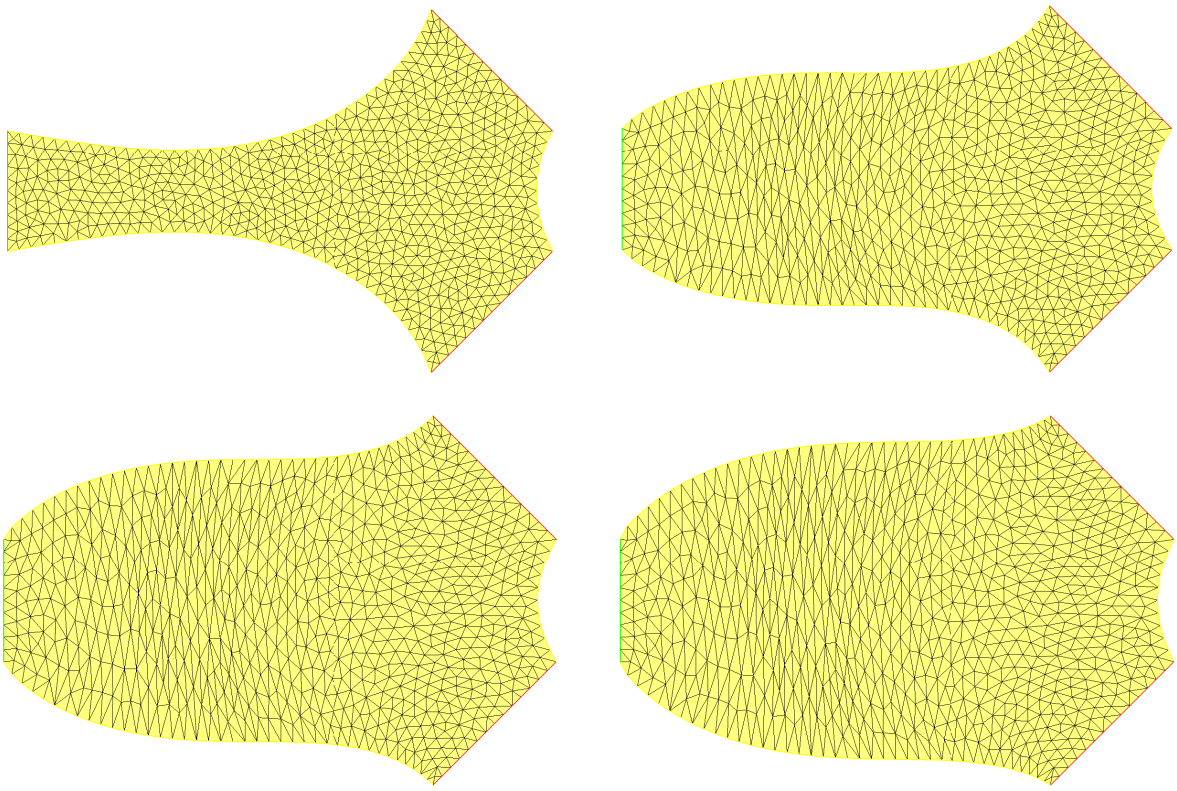

FIGURE 10. *From left to right, top to bottom, successive shapes* Ω^n *at iterations* $n = 0, 100, 250, 500$ *in the energy dissipation example in a ramified structure with perimeter constraint of Section 4.3.*

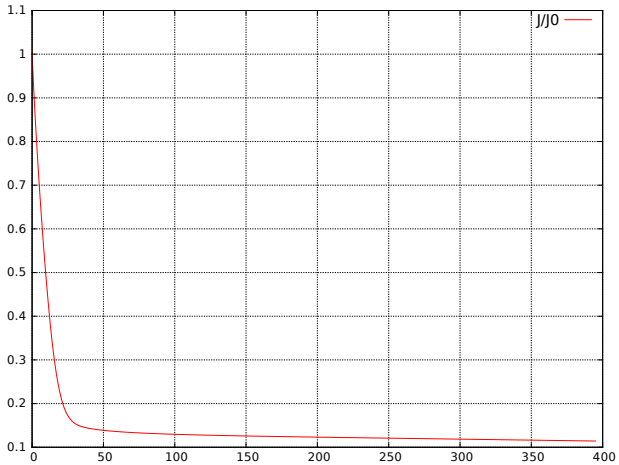

FIGURE 11. *Convergence history of* $J(\Omega)$ *in the least-square criterion minimization example of Section 4.4.*

where the boundary integral has vanished thanks to the boundary conditions satisfies by $(\mathbf{u}', p')$ and $(\mathbf{v}_d, q_d)$. Likewise, multiplying the first equation in (2.12) by $\mathbf{u}'$ and integrating by parts, we obtain:

$$
\begin{aligned}
0 &= \int_\Omega \left(-\nu\Delta\mathbf{v}_d + (\nabla\mathbf{u})^T\mathbf{v} - (\nabla\mathbf{v}_d)\mathbf{u} + \nabla q_d\right)\cdot\mathbf{u}'\,dx \\
&= \int_\Omega \left(2\nu\, e(\mathbf{u}') : e(\mathbf{v}_d) + (\nabla\mathbf{u})\mathbf{u}'\cdot\mathbf{v}_d - (\nabla\mathbf{v}_d)\mathbf{u}\cdot\mathbf{u}'\right)dx - \int_{\partial\Omega}\sigma(\mathbf{v}_d, q_d)\mathbf{n}\cdot\mathbf{u}'\,ds. \qquad \text{(A.4)}
\end{aligned}
$$

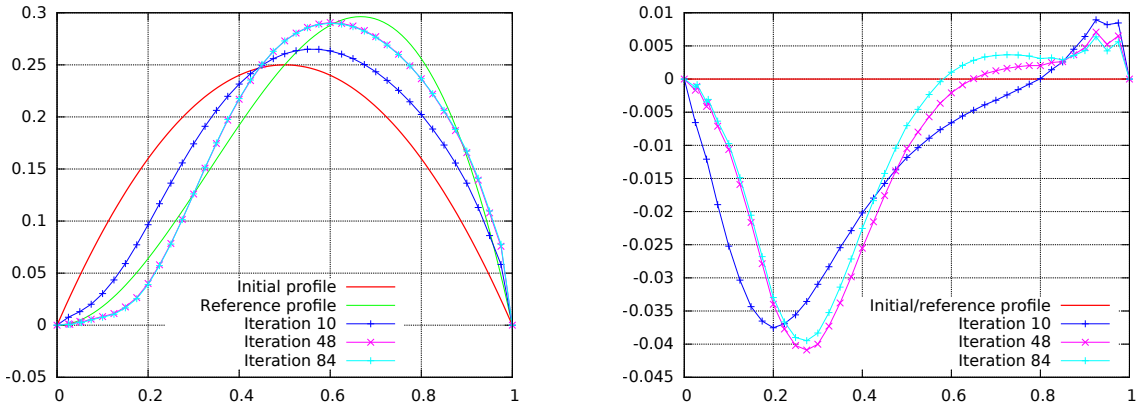

FIGURE 12. *One-dimensional profiles of (left)* u_1 *and (right)* u_2 *on* Γ_{out} *at several stages in the example of Section 4.4.*

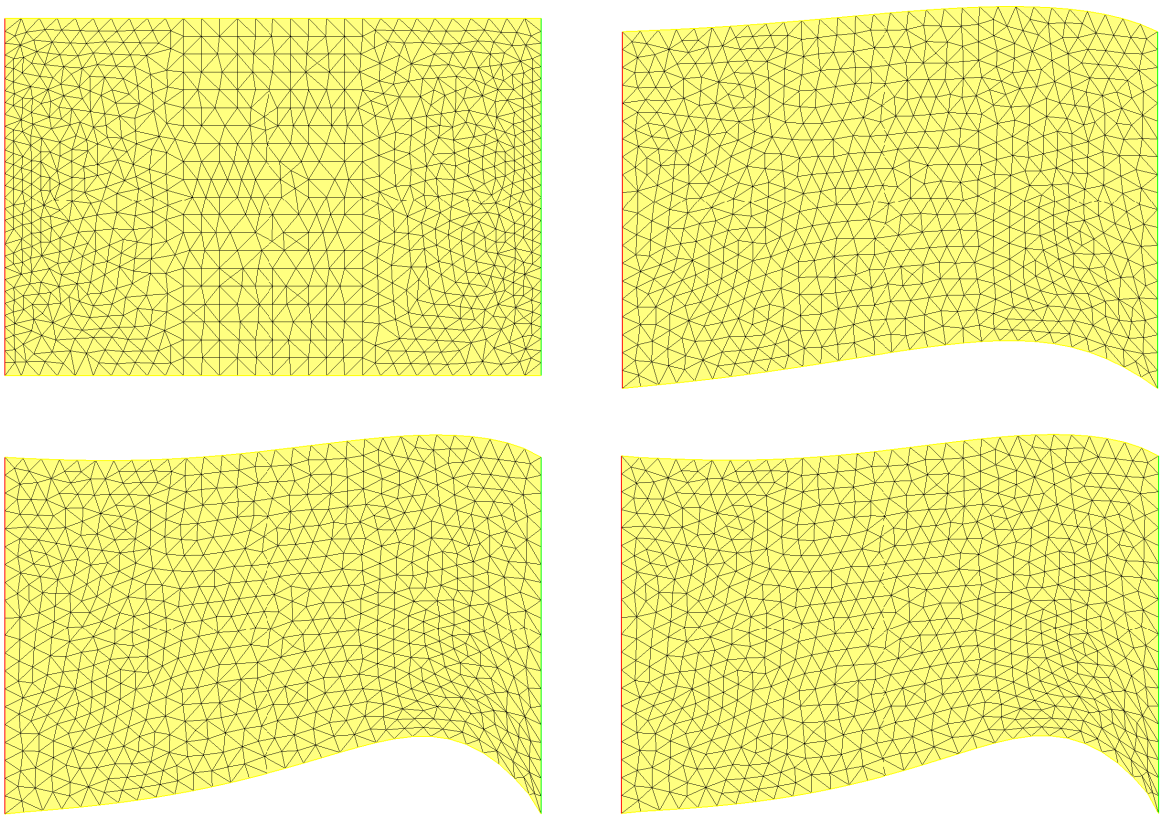

FIGURE 13. *From left to right, top to bottom, successive shapes* Ω^n *at iterations* $n = 0, 10, 48, 84$ *in the least-square criterion minimization example of Section 4.4.*

Combining equations (A.3) and (A.4) leads to:

$$-\int_\Omega ((\nabla \mathbf{u}')\mathbf{u}\cdot\mathbf{v}_d + (\nabla \mathbf{v}_d)\mathbf{u}\cdot\mathbf{u}')\,dx = \int_{\partial\Omega} \sigma(\mathbf{v}_d, q_d)\mathbf{n}\,ds. \tag{A.5}$$

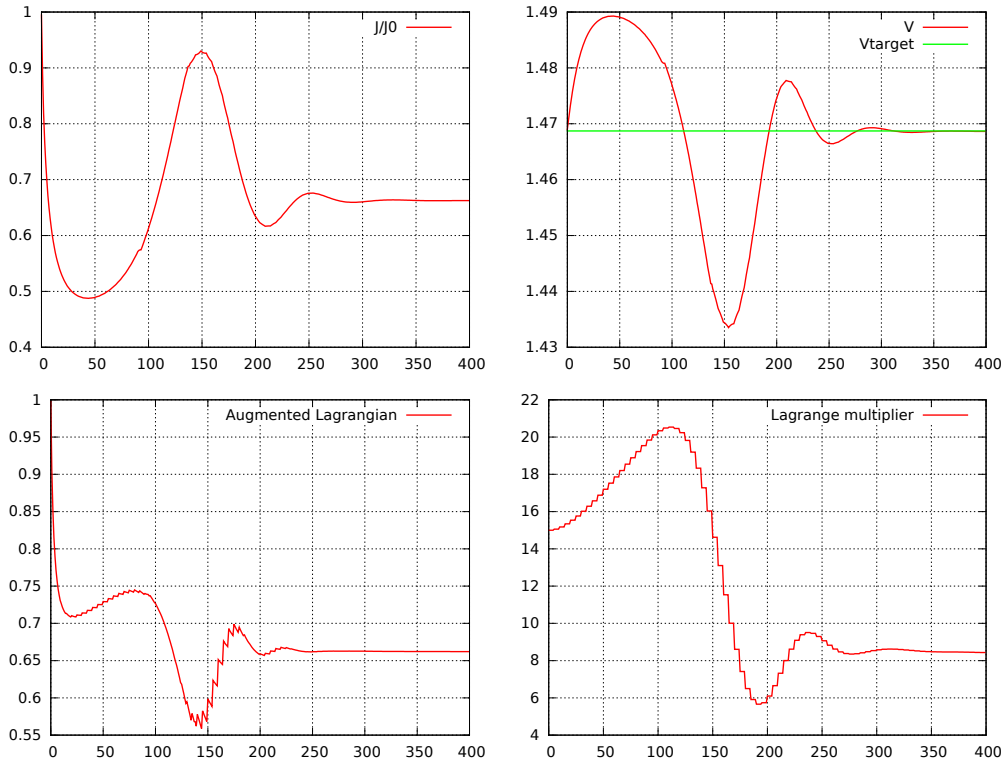

FIGURE 14. *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in the dissipated energy minimization example of Section 4.5*

Now using the identity

$$\int_\Omega (\nabla \mathbf{v}_d)\cdot\mathbf{u}\cdot\mathbf{u}'\,dx = \int_{\partial\Omega}(\mathbf{v}_d\cdot\mathbf{u}')(\mathbf{u}\cdot\mathbf{n})\,ds - \int_\Omega (\nabla\mathbf{u}')\mathbf{u}\cdot\mathbf{v}_d\,dx, \tag{A.6}$$

which again follows from integration by parts, Equation (A.5) rewrites:

$$\int_{\partial\Omega}(\sigma(\mathbf{v}_d,q_d)\mathbf{n}\cdot\mathbf{u}' + (\mathbf{u}'\cdot\mathbf{v}_d)(\mathbf{u}\cdot\mathbf{n}))\,ds = 0. \tag{A.7}$$

Eventually, taking into account the boundary conditions in the systems (2.1), (2.10) and (A.1) yields:

$$\begin{aligned} D'(\Omega)(\boldsymbol{\theta}) = \int_{\Gamma_{\mathrm{out}}}(\mathbf{u}-\mathbf{u}_{\mathrm{ref}})\cdot\mathbf{u}'\,ds &= \int_{\Gamma_{\mathrm{out}}}(\sigma(\mathbf{v}_d,q_d)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_d)\cdot\mathbf{u}'\,ds,\\ &= -\int_\Gamma(\sigma(\mathbf{v}_d,q_d)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_d)\cdot\mathbf{u}'\,ds,\\ &= \int_\Gamma(\sigma(\mathbf{v}_d,q_d)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_d)\cdot\frac{\partial\mathbf{u}}{\partial n}\,\boldsymbol{\theta}\cdot n\,ds. \end{aligned} \tag{A.8}$$

We now use the boundary conditions $\mathbf{u} = 0$ and $\mathbf{v}_d = 0$ on Γ to simplify this last expression. For any tangential vector field $\boldsymbol{\tau}:\Gamma\to\mathbb{R}^d$ to Γ, they imply that $\frac{\partial\mathbf{u}}{\partial\boldsymbol{\tau}} = 0$, and so, using that $\mathrm{div}(\mathbf{u}) = 0$,

$$\frac{\partial\mathbf{u}}{\partial n}\cdot\mathbf{n} = 0; \tag{A.9}$$

the same relation holds for $\mathbf{v}_d$. Hence (A.8) rewrites:

$$D'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma 2\nu\, e(\mathbf{v}_d)\mathbf{n}\cdot\frac{\partial\mathbf{u}}{\partial n}\,\boldsymbol{\theta}\cdot\mathbf{n}\,ds.$$

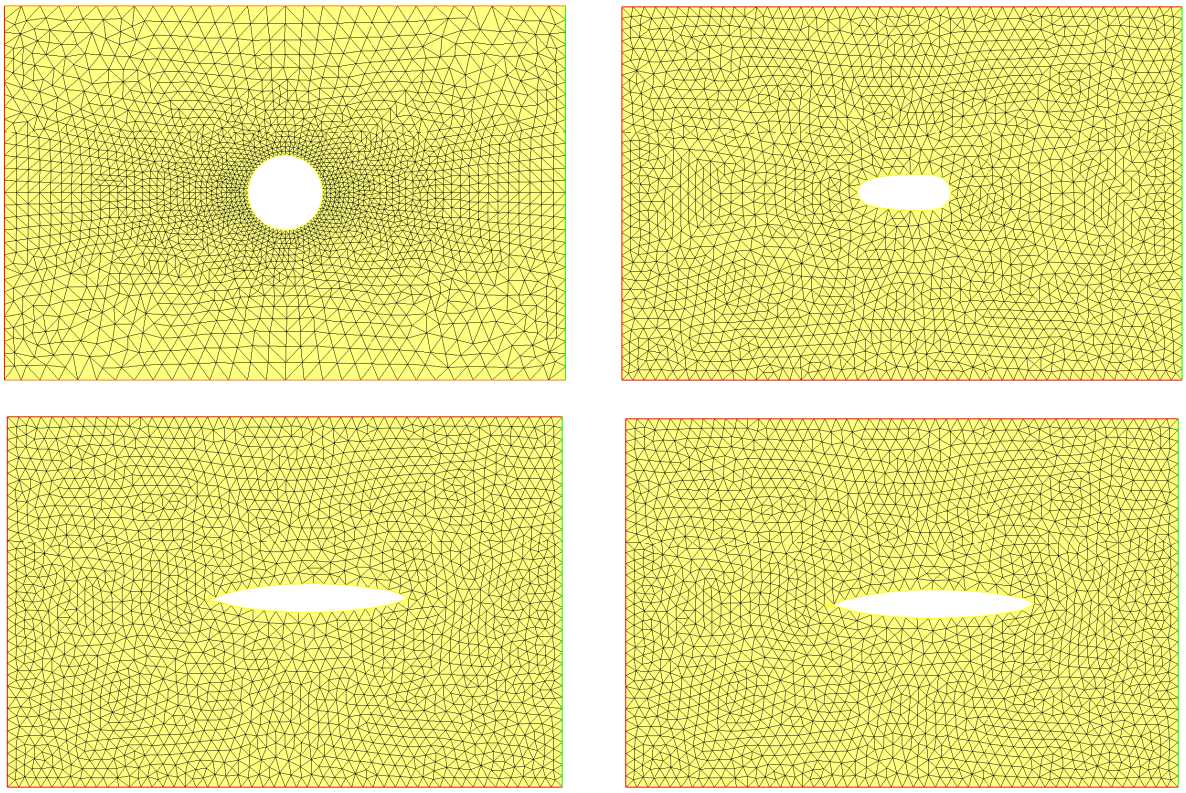

FIGURE 15. *From left to right, top to bottom, successive shapes* Ω^n *at iterations* $n = 5,\ 100,\ 650$ *in the dissipated energy minimization example of Section 4.5.*

After a few algebraic manipulations based again on (A.9), we eventually obtain:

$$D'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma 2\nu\, e(\mathbf{u}) : e(\mathbf{v}_d)(\boldsymbol{\theta}\cdot\mathbf{n})\, ds, \tag{A.10}$$

which is the desired result, and terminates the proof of Theorem 2.

REFERENCES

[1] N. AAGE, T. H. POULSEN, A. GERSBORG-HANSEN, AND O. SIGMUND, *Topology optimization of large scale stokes flow problems*, Struct. Multidisc. Optim., 35 (2008), pp. 175–180.

[2] F. ABRAHAM, M. BEHR, AND M. HEINKENSCHLOSS, *Shape optimization in unsteady blood flow: A numerical study of non-newtonian effects*, Computer Methods in Biomechanics and Biomedical Engineering, 8:3 (2005).

[3] V. AGOSHKOV, A. QUARTERONI, AND G. ROZZA, *Shape design in aorto-coronaric bypass anastomoses using perturbation theory*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 367–384.

[4] G. ALLAIRE, *Conception optimale de structures*, vol. 58, Springer, 2007.

[5] 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.

[6] G. ALLAIRE AND O. PANTZ, *Structural optimization with freefem++*, Structural and Multidisciplinary Optimization, 32 (2006), pp. 173–181.

[7] S. AMSTUTZ, *The topological asymptotic for the navier-stokes equations*, ESAIM: Control, Optimisation and Calculus of Variations, 11 (2005), pp. 401–425.

[8] M. BADRA, F. CAUBET, AND M. DAMBRINE, *Detecting an obstacle immersed in a fluid by shape optimization methods*, unpublished, (2011).

[9] T. J. BAKER, *Mesh movement and metamorphosis*, Eng. Comput., 18 (2002), pp. 188–198.

[10] M. P. BENDSOE AND O. SIGMUND, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.

[11] M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the stokes problem in the primitive variables*, Numerische Mathematik, 33 (1979), pp. 211–224.

[12] M. Bergounioux and Y. Privat, *Shape optimization with Stokes constraints over the set of axisymmetric domains*, SIAM J. Control Optim., 51 (2013), pp. 599–628.

[13] T. Borrvall and J. Petersson, *Topology optimization of fluids in stokes flow*, Int. J. Numer. Meth. Fluids, 41 (2003), pp. 77–107.

[14] C.-H. Bruneau, F. Chantalat, A. Iollo, B. Jordi, and I. Mortazavi, *Modelling and shape optimization of an actuator*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 1143–1151.

[15] D. Bucur and G. Buttazzo, *Variational methods in some shape optimization problems*, Appunti dei Corsi Tenuti da Docenti della Scuola. [Notes of Courses Given by Teachers at the School], Scuola Normale Superiore, Pisa, 2002.

[16] M. Burger, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.

[17] H. W. Carlson and W. D. Middleton, *A numerical method for the design of camber surfaces of supersonic wingswith arbitrary planforms*, NASA Technical report, (1964).

[18] V. Challis and J. Guest, *Level set topology optimization of fluids in stokes flow*, Int. J. Numer. Meth. Engng, 79 (2009), pp. 1284–1308.

[19] J.-H. Choi, K.-Y. Kim, and D.-S. Chung, *Numerical optimization for design of an automotive cooling fan*, tech. rep., SAE Technical Paper, 1997.

[20] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, Society for Industrial and Applied Mathematics, 2002.

[21] H. Çlabuk and V. Modi, *Optimum plane diffusers in laminar flow*, Journal of Fluid Mechanics, 237 (1992), pp. 373–393.

[22] 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.

[23] X. D. de La Sablonière, B. Mauroy, and Y. Privat, *Shape minimization of the dissipated energy in dyadic trees*, Discrete Contin. Dyn. Syst. Ser. B, 16 (2011), pp. 767–799.

[24] E. H. de Vasconcelos Segundo, A. L. Amoroso, V. C. Mariani, and L. dos Santos Coelho, *Economic optimization design for shell-and-tube heat exchangers by a tsallis differential evolution*, Applied Thermal Engineering, 111 (2017), pp. 143 – 151.

[25] C. Dobrzynski and P. Frey, *Anisotropic delaunay mesh adaptation for unsteady simulations*, Proc. 17th Int. Meshing Roundtable, (2008).

[26] G. Dogan, P. Morin, R. H. Nochetto, and M. Verani, *Discrete gradient flows for shape optimization and applications*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 3898–3914.

[27] X.-B. Duan, Y.-C. Ma, and R. Zhang, *Shape-topology optimization for navier-stokes problem using variational level set method*, Journal of Computational and Applied Mathematics, 222 (2008), pp. 487–499.

[28] A. Ern and J.-L. Guermond, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media, 2013.

[29] A. Evgrafov, *Topology optimization of slightly compressible fluids*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 86 (2006), pp. 46–62.

[30] P. Frey and P.-L. George, *Mesh generation, application to Finite Elements*, Wiley & Sons, 2008.

[31] H. Garcke, C. Hecht, M. Hinze, and C. Kahle, *Numerical approximation of phase field based shape and topology optimization for fluids*, SIAM Journal on Scientific Computing, 37 (2015), pp. A1846–A1871.

[32] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber, *Topology optimization of channel flow problems*, Structural and Multidisciplinary Optimization, 30 (2005), pp. 181–192.

[33] V. Girault and P.-A. Raviart, *Finite Element methods for Navier-Stokes Equations*, Springer Verlag, 1986.

[34] J. Guest and J. Prévost, *Topology optimization of creeping fluid flows using a darcystokes finite element*, Int. J. Numer. Meth. Engng, 66 (2006), pp. 461–484.

[35] M. D. Gunzburger, *Perspectives in flow control and optimization*, vol. 5, Siam, 2003.

[36] F. Hecht, O. Pironneau, A. Le Hyaric, and K. Ohtsuka, *Freefem++ manual*, 2005.

[37] A. Henrot and M. Pierre, *Variation et optimisation de formes*, vol. 48, Springer-Verlag Berlin Heidelberg, 2005.

[38] A. Henrot and Y. Privat, *Une conduite cylindrique n'est pas optimale pour minimiser l'énergie dissipée par un fluide*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 1057–1061.

[39] ———, *What is the optimal shape of a pipe?*, Arch. Ration. Mech. Anal., 196 (2010), pp. 281–302.

[40] R. M. Hicks and P. A. Henne, *Wing design by numerical optimization*, Journal of Aircraft, 15 (1978), pp. 407–412.

[41] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats, *An assessment of airfoil design by numerical optimization*, (1974).

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

[43] J. L. Lions, *Optimal control of systems governed by partial differential equations*, vol. 170, Springer Verlag, 1971.

[44] A. Litman, D. Lesselier, and F. Santosa, *Reconstruction of a two-dimensional binary obstacle by controlled evolution of a level-set*, Inverse Problems, 14 (1998), p. 685.

[45] B. Mohammadi and O. Pironneau, *Shape optimization in fluid mechanics*, Annu. Rev. Fluid Mech., 36 (2004), pp. 255–279.

[46] B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Oxford University Press, 2010.

[47] B. Munson, A. Rothmayer, T. Okiishi, and W. Huebsch, *Fundamentals of Fluid Mechanics*, John Wiley & Sons, Inc., 7th edition ed., 2013.

[48] B. R. Munson, T. H. Okiishi, W. W. Huebsch, and A. P. Rothmayer, *Fundamentals of fluid mechanics, 7th edition*, Wiley & Sons, 2012.
[49] F. Murat and J. Simon, *Sur le contrôle par un domaine géométrique*, Technical report RR-76005, (1976).
[50] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2006.
[51] A. A. Novotny and J. Sokołowski, *Topological derivatives in shape optimization*, Springer Science & Business Media, 2012.
[52] 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.
[53] V. Patel and R. Rao, *Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique*, Applied Thermal Engineering, 30 (2010), pp. 1417 – 1425.
[54] G. Pingen, A. Evgrafov, and K. Maute, *Topology optimization of flow domains using the lattice boltzmann method*, Structural and Multidisciplinary Optimization, 34 (2007), pp. 507–524.
[55] O. Pironneau, *On optimum design in fluid mechanics*, Journal of Fluid Mechanics, 64 (1974), pp. 97–110.
[56] ———, *Optimal shape design for elliptic systems*, Springer Science & Business Media, 2012.
[57] J. A. Sethian and A. Wiegmann, *Structural boundary design via level set and immersed interface methods*, Journal of computational physics, 163 (2000), pp. 489–528.
[58] O. Sigmund, *A 99 line topology optimization code written in matlab*, Structural and multidisciplinary optimization, 21 (2001), pp. 120–127.
[59] J. Sokołowski and J.-P. Zolésio, *Introduction to shape optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.
[60] R. Temam, *Navier-Stokes Equation: Theory and Numerical Analysis*, North Holland, 1977.
[61] 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.
[62] S. Zhou and Q. Li, *A variational level set method for the topology optimization of steady-state navier-stokes flow*, J. Comput. Phys., 227 (2008), pp. 10178–10195.