Geometry optimization applied to incompressible fluid mechanics

Florian Omnès

Sorbonne Université

Laboratoire Jacques-Louis Lions - UMR 7598

Optimisation géométrique appliquée à la mécanique des fluides incompressibles

Thèse de doctorat

présentée et soutenue publiquement le 7 novembre 2018
pour l'obtention du

Doctorat de Sorbonne Université
Spécialité Mathématiques appliquées

par

Florian Omnès

Composition du jury

Rapporteurs :

Bijan MOHAMMADI	Professeur à l'Université de Montpellier
Jérôme FEHRENBACH	MCF à l'Institut de Mathématiques de Toulouse

Examinateurs :

Samuel AMSTUTZ	MCF à l'Université d'Avignon
François DETCHEVERRY	CR CNRS à l'Université Lyon 1
Laurent DUMAS	Professeur à l'Université Versailles Saint-Quentin-en-Yvelines
Emmanuel TRELAT	Professeur à Sorbonne Université

Invités :

Matthieu BONNIVARD	MCF à l'Université Paris-Diderot
Marcela SZOPOS	Professeur à l'Université Paris Descartes

Directeurs de thèse :

Pascal FREY	Professeur à Sorbonne Université
Yannick PRIVAT	Professeur à l'Université de Strasbourg

École doctorale de Sciences Mathématiques
de Paris Centre ED 386

Faculté de Mathématiques
UFR 929

Florian Omnès :

Sorbonne Université, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

Sorbonne Université, Institut des sciences du calcul et des données, ISCD, FED-3, F-75005 Paris, France

Adresse électronique: florian.omnes@sorbonne-universite.fr

Remerciements

Tout d'abord, je voudrais remercier mes directeurs de thèse. Ils ont su se montrer disponibles tout au long de ma thèse, m'apporter leurs connaissances scientifiques et rédactionnelles et me transmettre leur goût pour les mathématiques appliquées. Je remercie également Matthieu Bonnivard et Charles Dapogny avec qui j'ai eu le plaisir de collaborer pendant ma thèse.

Jérôme Fehrenbach et Bijan Mohammadi m'ont fait l'honneur de rapporter cette thèse. Je les remercie pour avoir porté un intérêt tout particulier à mon travail et pour leurs remarques qui m'ont permis d'améliorer ce manuscrit. Je remercie également Samuel Amstutz, François Detcheverry, Laurent Dumas, Emannuel Trélat, Matthieu Bonnivard et Marcela Szopos pour leur présence dans le jury.

Je remercie le Laboratoire Jacques-Louis Lions pour m'avoir accueilli pendant ces trois années, particulièrement Salima, Catherine, Malika et Khashayar pour leur professionnalisme et leur bonne humeur. Je remercie les doctorants du bureau : Ludovic, Hugo, Igor, Guillaume, Malik, Chien Yu, et les autres pour toutes les discussions passionnantes que avons partagées. Je remercie également Ludovic, Shuyang, Amaury, Jean et Idriss pour avoir partagé avec moi la responsabilité du Groupe de Travail des Thésards.

Je remercie mes amis Brieuc, Alexis et Vincent pour m'avoir infligé de nombreuses défaites vidéoludiques, permis de découvrir entre autres la cuisine éthiopienne et congolaise, et accueilli sur l'île de Groix.

Enfin, ma famille m'a apporté un soutien indéfectible tout au long de mes études puis pendant ma thèse. La présence de ma

femme Shuyang à mes côtés et ses conseils m'ont permis de dépasser tous les moments de doute. A tous, merci.

Contents

1 Introduction générale **1**

1.1 Optimisation géométrique pour un fluide de Navier-Stokes 4

1.1.1 Présentation du problème 4

1.1.2 Objectifs du Chapitre 2 6

1.1.3 Description de la méthode numérique 7

1.1.4 Difficultés numériques 8

1.1.5 Avantages et limitations de l'algorithme . . . 10

1.1.6 Exemple : énergie minimisée par une structure ramifiée 10

1.2 Modélisation et optimisation géométrique d'une aquaporine 11

1.2.1 Présentation du problème 11

1.2.2 Objectifs du Chapitre 3 14

1.2.3 Motivations applicatives 15

1.2.4 Méthode numérique d'optimisation de la forme d'une aquaporine 16

2 Geometry optimization of a Navier-Stokes flow **19**

2.1 State of the art 20

2.1.1 Geometry optimization 20

2.1.2 Problem discretization 21

2.1.3 Sensitivity analysis : adjoint methods 25

2.1.4 Topology optimization 25

2.1.5 Homogenization methods 28

2.1.6 Theoretical issues 30

2.2 Statement of the shape optimization problem 31
2.3 Shape sensitivity analysis using Hadamard's boundary variation method 33
2.4 Numerical methods 38
2.4.1 Description of the numerical setting and outline of the algorithm 39
2.4.2 Numerical resolution of the Navier-Stokes equations 40
2.4.3 The augmented Lagrangian algorithm for equality-constrained problems 44
2.4.4 Mesh-related issues 46
2.4.5 Extension-regularization of the shape gradient 48
2.4.6 Calculation of the curvature 52
2.4.7 Algorithmic description of the implemented method 54
2.5 Numerical illustrations 55
2.5.1 Minimization of the dissipated energy in a bending pipe 57
2.5.2 Minimization of the dissipated energy in a ramified structure with volume constraint . 57
2.5.3 Minimization of the dissipated energy in a ramified structure with perimeter constraint 60
2.5.4 Squared vorticity in a ramified structure . . 63
2.5.5 Minimization of the discrepancy with a reference velocity profile 64
2.5.6 Energy dissipation around an obstacle . . . 70
2.5.7 Effect of the mesh refinement on numerical results 70
2.6 Conclusion and perspectives 76

3 Geometry optimization of an hourglass-shaped aquaporin 77
3.1 State of the art 78
3.1.1 Physical modelling of aquaporins 78

3.1.2 Numerical simulation of fluid flow in an aquaporin . . . 78
3.2 Introduction and modeling of the problem . . . 81
3.2.1 Geometry and fluid model . . . 81
3.2.2 The shape optimization problem . . . 91
3.3 Analysis of the shape optimization problem . . . 92
3.3.1 Existence issues . . . 92
3.3.2 Computation of the shape derivative of J . . . 98
3.4 Numerical methods and algorithms . . . 106
3.4.1 Choice of parameters . . . 108
3.4.2 Finding the optimal angle (step 1 of the algorithm) . . . 109
3.4.3 Optimizing the shape of Γ_1 (step 2 of the algorithm) . . . 110
3.4.4 Numerical results . . . 114
3.5 Further comments and conclusion . . . 118
3.6 Shape-optimization of an axisymmetric aquaporin . . . 120
3.6.1 Axisymmetric Stokes equations . . . 120
3.6.2 Energy formulation of the Stokes problem . . . 121
3.6.3 Result of symmetry . . . 121
3.6.4 Variational formulation of the problem in a meridian . . . 122
3.6.5 Shape derivative of the energy functional . . . 124
3.6.6 First numerical results . . . 125

4 Conclusion and perspectives **129**

Appendices **133**
A Computation of shape derivatives . . . 133
A.1 Shape derivative of the functional E . . . 134
A.2 Shape derivative of the functional D . . . 136
B Practical implementation of the shape optimization algorithm . . . 137
B.1 Organization of the repository and of the program . . . 138

B.2 Main parameters 138
B.3 Main macros 139
B.4 Definition of the geometry and of the Finite Element setting 139
B.5 Practical calculation of the mean curvature . 142
B.6 Resolution of the flow equations 142
B.7 Calculation of the objective function and of the shape derivative 144
B.8 Main optimization loop : gradient descent with line search 146
C Proof of the axisymmetry result on a fixed domain 149

List of Figures

1.1 *Profil initial (gauche) et optimisé (droite) d'un avion supersonique. Les couleurs représentent la contribution locale à la traînée, qui est le critère à optimiser dans ce cas. Crédit J. Reuthers, J.-J. Alonso, M.-J. Rimlinger and A. Jameson [124]* 2

1.2 *Exemple d'optimisation empirique : le déflecteur d'air installé au-dessus de la cabine permet de dévier l'écoulement, ce qui permet de diminuer la traînée associée. (photo placée dans le domaine public)* 3

1.3 *Illustration d'un domaine* Ω *avec entrée* Γ_{in}*, sortie* Γ_{out} *fixées. La frontière* Γ *est modifiée pour minimiser le critère choisi.* 5

1.4 *Schéma du cas-test "structure ramifiée", avec 4 entrées* Γ_{in} *et une sortie* Γ_{out} 11

1.5 *Formes successives* Ω^n *aux itérations* $n = 0, 5, 240, 1000$ *pour le cas de l'optimisation de la forme d'une structure ramifiée pour le critère "Energie dissipée" Section 2.5.2.* 12

1.6 *Graphiques de convergence (de gauche à droite, de haut en bas)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *et* ℓ^n *dans le cas de l'optimisation d'une structure ramifiée le critère "Energie dissipée" Section 2.5.2.* 13

1.7 *Schéma d'une aquaporine. Les molécules d'eau sont transportées dans un canal à travers la membrane cellulaire, par une aquapore en forme de sablier. Courtoisie de François Detcheverry (ILM, Lyon 1).* . . . 16

2.1 *Example of shape optimization of a fluid distributor. The flow is modelled using turbulent Lattice-Boltzmann equations at Re 100-30000. Fluid and solid cells at the boundary are iteratively swapped using a heuristic criterion until a residual falls under a fixed threshold. Credits : L. Wang, Y. Fan and Lingai Luo [136]* 22

2.2 *In this example, the level-set method is used to maximize the ferromagnetic repulsive force between a fixed object (bottom) and a deformable ferromagnetic material (top). Credits : H. W. Kuhn and A. W. Tucker, [99].* 24

2.3 *Evolution of a solid obstacle immersed in a cubic domain of Stokes fluid during the topology optimization procedure. A relaxation method is used to minimize the energy dissipated by the fluid. Credits : T. Borrvall and J. Petersson. [30]* 26

2.4 *Evolution of a solid obstacle immersed in a cubic domain of fluid during the minimization procedure. A 3D level-set method is used to minimize the energy dissipated by the fluid. Credits : S. Zhou and Q. Li [139]* 27

2.5 *Evolution of a solid obstacle immersed in a rectangular domain of fluid during the minimization procedure. A phase-field method is used to minimize the energy dissipated by the fluid. Credits : H. Garcke, C. Hecht, M. Hinze and C. Kahle [68]* 29

2.6 *3D shape with a lower dissipated energy than the cylinder. Credits : A. Henrot and Y. Privat [83]* . . 30

2.7 *Example of a variation $\Omega_{\boldsymbol{\theta}}$ of a shape Ω. First iteration of the bending pipe example, Section 2.5.1. The initial domain is represented in blue, the deformed one is in green. Note : the arrows are not to scale.* . 35

2.8 *Examples of a mesh with overlapping triangles (in gray, left); a mesh with non overlapping, yet non conforming triangles (in gray, middle); a computational mesh, right.* 39

2.9 *Examples of (a) a mesh getting very stretched (gray elements); (b) a mesh developing overlaps (red elements) in the course of its deformation.* 47

2.10 *Illustrations of the remeshing operations described in Section 2.4.4: (a) splitting of a 'long' edge; (b) collapse of the two endpoints of a 'short' edge; (c) swap of the connectivities of a configuration of two ill-shapes triangles; (d) relocation of one vertex.* . . 49

2.11 *Estimation of the tangent and normal vectors to $\partial\Omega$ from the data of a triangular mesh.* 53

2.12 *Synthetic description of the optimization loop* 54

2.13 *Settings of the five test cases discussed in Section 2.5; (a) the bend, discussed in Section 2.5.1, (b) the ramified structure of Sections 2.5.2 and 2.5.3, (c) the straight pipe with one inlet, one outlet where a least-square criterion is considered, as studied in Section 2.5.5, (d) the dissipated energy minimization example of Section 2.5.6.* 55

2.14 *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 bend optimization example of Section 2.5.1* 58

2.15 *Intermediate shapes Ω^n obtained in the bend optimization example of Section 2.5.1 at iterations (from top to bottom) $n = 0, 5, 100$ and 500.* 59

2.16 *From 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 2.5.2.* 60

2.17 *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 2.5.2.* 61

2.18 *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 2.5.3* 62

2.19 *From 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 2.5.3.* 63

2.20 *Left upper corner of intermediate shapes* Ω^n *obtained in the ramified structure example of Section 2.5.3 at iterations (from top to bottom)* $n = 15, 60$ *and* 130. 64

2.21 *Intermediate shapes* Ω^n *obtained in the ramified structure example of Section 2.5.4 at iterations (from top to bottom)* $n = 200, 300$ *and* 474. 65

2.22 *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Per}(\Omega)$, $\mathscr{L}(\Omega,\ell,b)$ *and* ℓ^n *in squared vorticity minimization in a ramified structure example of Section 2.5.4* 66

2.23 *Stopping criterion in the minimization in a ramified structure example of Section 2.5.4* 67

2.24 *Convergence history of* $J(\Omega)$ *in the least-square criterion minimization example of Section 2.5.5.* . . . 67

2.25 *One-dimensional profiles of (top)* u_1 *and (bottom)* u_2 *on* Γ_{out} *at several stages in the example of Section 2.5.5.* 68

2.26 *From top left to bottom right, successive shapes* Ω^n *at iterations* $n = 0, 6, 100$ *and reference shape in the least-square criterion minimization example of Section 2.5.5.* 68

2.27 *Boundary of the reference shape and shape at convergence in the least-square criterion minimization example of Section 2.5.5.* 69
2.28 *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 2.5.6* . 71
2.29 *From top to bottom, successive shapes* Ω^n *at iterations* $n = 5,\ 100,\ 650$ *in the dissipated energy minimization example of Section 2.5.6.* 72
2.30 *Boundaries of converged domains with different mesh sizes in the bend optimization example of Section 2.5.1. Bottom : detail on the gray area featured in the top figure.* 73
2.31 *Evolution of the cost functional* J *for mesh sizes* $h = 1/30$, $h = 1/50$ *and* $h = 1/70$. 74
2.32 *From top to bottom : Initial (left) and final (right) mesh for mesh sizes* $h = 1/30$, $h = 1/50$ *and* $h = 1/70$. 75

3.1 Illustration of the slip-length b. The linear extrapolation of the velocity profile vanishes at depth b inside the solid. 79
3.2 *The domain* Ω *(two reservoirs connected by an hourglass shaped channel).* 81
3.3 *The external, rectangular box* D *contains the domain* Ω *(in blue), completed with two lateral sub-regions (in pink).* 94
3.4 *Computational domains for* $\theta = 0.1, 0.2$ *and* 0.4*, from top to bottom.* 109
3.5 *Criterion* $J(\Omega)$ *w.r.t. the angle parameter in Test-cases 1 and 2 (see Table 3.1).* 110
3.6 *From top to bottom : mesh at iterations 0, 45 and 85 for Test-case 1.* 115
3.7 *From top to bottom : mesh at iterations 0, 130 and 245 for Test-case 2.* 115

3.8 *Test-case 1. Left: evolution of the cost functional J, and its sub-parts J_d, J_f over iterations, relatively to the initial value J_0 of J. Right: evolution of the stopping criterion.* 116
3.9 *Test-case 2. Left: evolution of the cost functional J, and its sub-parts J_d, J_f over iterations, relatively to the initial value J_0 of J. Right: evolution of the stopping criterion.* 117
3.10 *Outline of regions. Each region is defined as the intersection of Ω with an infinite vertical strip $\{a < x_1 < b\}$. Regions 1, 2 and 3 (resp. 5, 6, 7) each take one third of the inlet (resp. outlet) cone width, region 4 is the whole central tube.* 117
3.11 *Evolution of each term J_i during step 2 of the optimization process. Test-case 1 on the left, 2 on the right.* 118
3.12 *Plot of the objective functional during the iterations, starting with two different initial values for θ. The red curve is obtained with θ^* extracted from Table 3.1, the green curve is associated with $\theta = 0.3$. The final value of J is 21% lower when the initial angle θ^* is chosen.* 120
3.13 Illustration of the meridian located at $\theta = 0$. 122
3.14 *Velocity and pressure profiles on an axisymmetric domain* 126
3.15 *Top : initial mesh. Bottom : displacement field associated to the first iteration* 127

Notations

$e(\mathbf{u})$ Strain rate tensor, defined by

$$e(\mathbf{u}) \coloneqq \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)$$

$\sigma(\mathbf{u}, p)$ Stress tensor. For an incompressible Newtonian fluid

$$\sigma(\mathbf{u}, p) \coloneqq 2\nu e(\mathbf{u}) - p\mathrm{I}$$

$(\mathbf{u} \cdot \nabla)\mathbf{u}$ Velocity convection term in the Navier-Stokes equations

$$((\mathbf{u} \cdot \nabla)\mathbf{v})_j \coloneqq \sum_{i=1}^{d} u_i \frac{\partial v_j}{\partial x_i} = (\nabla \mathbf{v})\mathbf{u}$$

$A : B$ Frobenius product of two tensors $A, B \in \mathcal{M}_n(\mathbb{R})$

$$A : B \coloneqq \mathrm{tr}(A^T B) = \sum_{i,j=1}^{n} a_{i,j} b_{i,j}$$

$[\mathbf{u}]_\tau$ Tangential part of a vector field on the boundary

$$[\mathbf{u}]_\tau = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n})\mathbf{n}$$

$\omega(\mathbf{u})$ Vorticity associated to the velocity field

$$\omega(\mathbf{u}) = \nabla \times \mathbf{u}$$

$S_n(\mathbb{R})$ Symmetric matrices of size n with real coefficients

Chapter 1

Introduction générale

L'optimisation de la forme d'objets au contact de fluides intervient dans de nombreux domaines : en ingénierie aéronautique, pour la conception de pièces intervenant dans des circuits fluides, ou plus récemment en biologie. L'amélioration de la forme d'une pièce mécanique pouvant être à l'origine de gains de performances importants, de nombreux travaux ont été consacrés à cette problématique, avant même l'avènement de la simulation numérique.

L'exemple le plus étudié dans ce champ d'application est sans doute celui de l'optimisation du profil d'une aile d'avion. Pour augmenter les performances de l'aile, on maximise la portance, ou bien on minimise la traînée, ou encore on maximise le rapport poussée/traînée. Plus récemment, des travaux ont permis d'optimiser la forme d'un avion entier, voir Figure 1.1.

La méthode d'optimisation de forme la plus ancienne, dite "d'essai et erreur" consiste à essayer plusieurs designs jusqu'à ce que l'expérimentateur soit satisfait, c'est-à-dire lorsque des performances jugées suffisantes sont atteintes en respectant les contraintes de départ. Ces contraintes peuvent être économiques (coût des matériaux, de l'assemblage) ou techniques (robustesse, encombrement). Cette recherche peut être guidée par des principes généraux d'aérodynamique, par exemple éviter les surfaces perpendiculaires à la trajectoire (voir figure 1.2). Les gains de performances peuvent être validés expérimentalement, par exemple dans une soufflerie.

Bien que la méthode par essais et erreur fournisse des gains

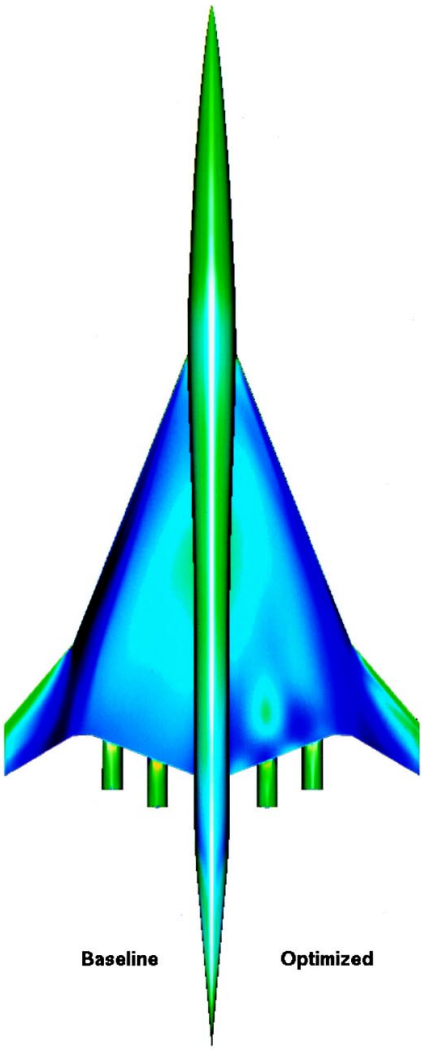


FIGURE 1.1 – *Profil initial (gauche) et optimisé (droite) d'un avion supersonique. Les couleurs représentent la contribution locale à la traînée, qui est le critère à optimiser dans ce cas. Crédit J. Reuthers, J.-J. Alonso, M.-J. Rimlinger and A. Jameson [124]*

importants, on peut espérer obtenir des gains supplémentaires à l'aide de méthodes plus systématiques, dont il sera question dans ce travail. Ces méthodes sont ici regroupées sous l'appellation générique "optimisation de forme", et recouvrent en réalité des approches très variées pour des objectifs communs.

Les exemples donnés ci-dessus concernent des écoulements dits à "haut Reynolds", qui font apparaître des phénomènes de turbu-

FIGURE 1.2 – *Exemple d'optimisation empirique : le déflecteur d'air installé au-dessus de la cabine permet de dévier l'écoulement, ce qui permet de diminuer la traînée associée. (photo placée dans le domaine public)*

Description	Eau dans un micro-canal en silicium [84]	Air dans les trois premières générations d'un poumon humain [104]
Viscosité cinématique ν à 20 °C ($m^2\,s^{-1}$)	1×10^{-6}	18.10^{-6}
Longueur caractéristique L - diamètre (m)	$15 \times 10^{-6} - 150 \times 10^{-6}$	1.34×10^{-2}
Vitesse caractéristique U (ms^{-1})	$1.2 - 10$	$0.27 - 2.161$
Nombre de Reynolds Re (sans dimension)	$34 - 1540$	$200 - 1600$

TABLE 1.1 – *Situations physiques d'écoulements à bas nombre de Reynolds*

lence qui induisent des difficultés d'ordre mathématique (existence et unicité de solutions, régularité par rapport au domaine) et numériques : coût de calcul, régularité par rapport au domaine. Ce travail se concentre exclusivement sur des écoulements laminaires ($Re < 1000$), modélisés par les équations de Stokes ou Navier-Stokes. De nombreux écoulements entrent dans ce cadre, quelques exemples sont donnés dans le tableau 1.1.

Le problème général consiste à minimiser une fonction régulière

J sous une contrainte géométrique de la forme $G(\Omega) = 0$:

$$\min_{\Omega, G(\Omega)=0} J(\Omega, \mathbf{u}_\Omega) \tag{1.1}$$

où $\mathbf{u}_\Omega$ est la solution d'un système de Stokes ou Navier-Stokes posé sur Ω (voir Figure 1.3).

$$\left\{ \begin{array}{rcll} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p & = & f & \text{in } \Omega, \\ \operatorname{div}(\mathbf{u}) & = & 0 & \text{in } \Omega, \\ \mathbf{u} & = & \mathbf{u}_{\text{in}} & \text{on } \Gamma_{\text{in}}, \\ \mathbf{u} & = & 0 & \text{on } \Gamma, \\ \sigma(\mathbf{u}, p)\mathbf{n} & = & 0 & \text{on } \Gamma_{\text{out}}. \end{array} \right. \tag{1.2}$$

1.1 Optimisation géométrique pour un fluide de Navier-Stokes

1.1.1 Présentation du problème

Dans cette partie, nous nous intéressons à des problèmes d'optimisation de forme du type

$$\min_{\Omega} J(\Omega) \tag{1.3}$$

où $J(\cdot)$ est une fonction dépendante de la solution de l'équation de Stokes ou Navier-Stokes incompressible, stationnaire en régime laminaire. Plus précisément, soit $(\mathbf{u}, p)$ la solution de (1.2). L'équation (1.2) peut se réécrire

$$F_\Omega(\mathbf{u}, p) = g_\Omega \tag{1.4}$$

pour une certaine fonction F_Ω , définie sur un espace fonctionnel $V(\Omega) \times Q(\Omega)$ et un second-membre g.

$$\begin{array}{lcl} \tilde{J} : \mathcal{O}_{\text{ad}} \times V(\Omega) \times Q(\Omega) & \to & \mathbb{R} \\ (\Omega, \mathbf{v}, q) & \mapsto & \tilde{J}(\Omega, \mathbf{v}, q) \end{array} \tag{1.5}$$

où $\mathcal{O}_{\text{ad}}$ représente l'ensemble des domaines admissibles. L'objectif principal de ce chapitre est d'exposer des méthodes numériques

permettant de résoudre, au moins localement, des problèmes de la forme

$$\begin{cases} \min \quad \tilde{J}(\Omega, \mathbf{v}, q). \\ \Omega \in \mathcal{O}_{\text{ad}} \\ F_\Omega(\mathbf{v}, q) = 0 \\ G(\Omega) = 0 \end{cases} \tag{1.6}$$

Nous nous placerons dans le cadre où, pour tout Ω dans $\mathcal{O}_{\text{ad}}$, le problème (1.4) admet une unique solution $(\mathbf{u}_\Omega, p_\Omega)$. Le problème (1.6) se réécrit alors

$$\begin{cases} \min \quad J(\Omega) \\ \Omega \in \mathcal{O}_{\text{ad}} \\ G(\Omega) = 0 \end{cases} \tag{1.7}$$

où

$$J(\Omega) = \tilde{J}(\Omega, \mathbf{u}_\Omega, p_\Omega).$$

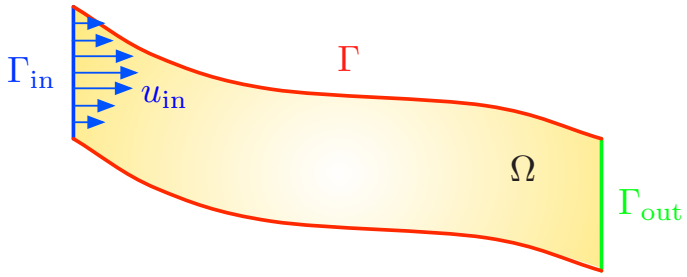


FIGURE 1.3 – *Illustration d'un domaine Ω avec entrée Γ_{in}, sortie Γ_{out} fixées. La frontière Γ est modifiée pour minimiser le critère choisi.*

L'ensemble des domaines admissibles est par exemple $\mathcal{O}_{\text{ad}}$ l'ensemble des domaines Lipschitz connexes dont la frontière contient Γ_{in} et Γ_{out}.

Dans chapitre 2 sont étudiées plusieurs fonctions coût dont l'énergie dissipée par le fluide par viscosité $E(\Omega)$, l'écart L^2 à un profil de vitesse référence sur un bord fixé $D(\Omega)$.

$$E(\Omega) = \int_\Omega \sigma(\mathbf{u}_\Omega, p) : e(\mathbf{u}_\Omega)\, dx = 2\nu \int_\Omega \|e(\mathbf{u}_\Omega)\|^2\, dx, \tag{1.8}$$

La contrainte $G(\Omega) = 0$ est une contrainte traduisant que le volume ou le périmètre de Ω est fixé, égal à une valeur cible. Bien sûr, d'autres contraintes géométriques pourraient également être utilisées.

1.1.2 Objectifs du Chapitre 2

Le chapitre 2 de la thèse a pour objet de présenter un cadre numérique souple et robuste permettant de traiter des problèmes d'optimisation géométrique du même type que (1.7). La méthode numérique permettant d'approcher une solution du problème (1.7) est construite sur plusieurs briques de base. L'un des objectifs de ce chapitre est de présenter en détail chacune de ces étapes.

Ce chapitre est dédié à la présentation d'une méthode robuste d'optimisation géométrique pour les fluides de Stokes et Navier-Stokes à bas Reynolds. Ce chapitre est l'occasion de rappeler les concepts théoriques nécessaires à la compréhension de l'algorithme, tels que la dérivée de forme (Section 2.3), la résolution numérique des équations de Navier-Stokes (Section 2.4.2) ou encore la méthode du lagrangien augmenté (Section 2.4.3).

Nous fournissons une implémentation disponible librement en ligne. Notre but, en sus de démontrer l'efficacité de l'algorithme sur des exemples, est d'en expliquer le fonctionnement. Étant donné que l'application de la méthode de variation de domaine à la mécanique des fluides n'est ni simple, ni standard en raison d'évidentes difficultés numériques, nous expliquons chaque étape de la boucle d'optimisation au sein d'un même document. En cela, nous espérons fournir un point d'entrée aux ingénieurs d'études et aux non-spécialistes désireux d'utiliser un cadre mathématique similaire pour traiter des problèmes d'optimisation de forme en mécanique des fluides. Les difficultés propres à l'implémentation sont exposées ainsi que des remèdes nécessaires au bon fonctionnement du programme : l'extension-régularisation de la dérivée de forme (Section 2.4.5) est un exemple. Le lecteur trouvera tous les ingrédients nécessaires à l'algorithme d'optimisation, avec une attention particulière sur les

difficultés numériques et une mise en relation du code informatique avec les notions mathématiques sous-jacentes.

1.1.3 Description de la méthode numérique

En général, le problème (1.7) n'admet pas de solution explicite. En effet, les équations de Stokes ou Navier-Stokes sur lesquelles s'appuient le problème d'optimisation de forme n'ont en général elles-mêmes pas de solution analytique. Pour cette raison, on utilise des méthodes numériques pour approcher les solutions de (1.7), lorsqu'elles existent.

L'approche que nous présentons est une approche "optimiser puis discrétiser" faisant intervenir un problème adjoint continu dans l'analyse de sensibilité. Les solutions des équations de Navier-Stokes et adjointes sont approchées grâce à la Méthode des Éléments Finis.

Les méthodes que nous présentons dans ce travail sont fondées sur la dérivée de forme du critère J, qui décrit la sensibilité de J par rapport à la position d'une partie du bord $\Gamma \subset \partial\Omega$. Plus précisément, étant donné un champ de déformation $\boldsymbol{\theta} \in W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)$, on étudie le développement au premier ordre de

$$\boldsymbol{\theta} \mapsto J((I + \boldsymbol{\theta})(\Omega))$$

Approximation de la solution de Navier-Stokes et du problème adjoint En deux dimensions, la frontière du domaine géométrique Ω est approchée par un polygone, puis triangulée. Sur cette triangulation, les équations de Navier-Stokes (1.2) et le problème adjoint continu, provenant de l'expression de la dérivée de forme, sont approchés par la méthode des éléments finis mixtes LBB-stables P2 (vitesse) / P1 (pression). La non-linéarité présente dans (1.2) est traitée par la méthode de Newton en résolvant une suite de problèmes d'Oseen, voir Section 2.4.2.

Descente de gradient et régularisation La méthode de descente de gradient suivant Ω s'appuie sur une méthode d'extension-régularisation, dont l'objectif est double. La dérivée de forme porte sur le bord variable Γ. Pour les applications numériques, et afin

éviter de recalculer une triangulation du domaine à chaque itération, le maillage est déformé en suivant le déplacement du bord. C'est l'objectif de la phase d'extension, qui étend l'information du bord Γ au maillage entier. L'étape d'extension-régularisation est détaillée dans la section 2.4.5.

En général, le gradient de forme numérique n'est pas régulier, en particulier près de coins et/ou de changement de conditions de bord, où peuvent apparaître des singularités. Si l'on déplace Γ suivant le gradient L^2 de forme, on fait bien décroître J, mais on obtient au bout de quelques itérations une frontière irrégulière. La géométrie obtenue ne présente pas d'intérêt physique car elle ne répond pas aux contraintes pratiques, par exemple de fabrication. Suivant l'approche présentée dans [51, 57], on définit le produit scalaire

$$\begin{aligned} \forall \boldsymbol{\theta}, \boldsymbol{\psi} \in V, \ \langle \boldsymbol{\theta}, \boldsymbol{\psi} \rangle_{V,\gamma} = \gamma \int_\Omega Ae(\boldsymbol{\theta}) : e(\boldsymbol{\psi}) \, dx \\ + (1-\gamma) \int_\Gamma \nabla_\Gamma \boldsymbol{\theta} \cdot \nabla_\Gamma \boldsymbol{\psi} \, ds. \end{aligned} \tag{1.9}$$

Où A est l'opérateur associé à la loi de Hooke

$$\begin{array}{lcl} A : S_n(\mathbb{R}) & \to & S_n(\mathbb{R}) \\ M & \mapsto & AM = 2\mu M + \lambda \operatorname{tr}(M) I \end{array}$$

La direction obtenue est plus régulière que le gradient $L^2(\Gamma)$ et c'est toujours une direction de descente.

Prise en compte des contraintes par lagrangien augmenté Afin de prendre en compte la contrainte $G(\Omega) = 0$, on introduit classiquement la fonction lagrangien augmenté, qui permet de prendre en compte la contrainte $G(\Omega) = 0$. On utilise le lagrangien augmenté pour ses propriétés de **régularisation** du problème dual. On peut montrer [65, Chapitre 3] que le lagrangien augmenté a les mêmes points-selle que le lagrangien original.

1.1.4 Difficultés numériques

Les étapes décrites dans les paragraphes précédents peuvent faire apparaître des difficultés d'ordre numérique, que nous présentons ici ainsi que quelques solutions pour les éviter.

Minima locaux Étant donné que l'algorithme utilisé repose sur une méthode de descente de gradient, les minima de $J(\Omega)$ obtenus ne sont en général pas globaux. Une initialisation différente du maillage et des paramètres ℓ et b pourrait conduire à un résultat différent.

Gradient inconsistant Dans ce travail, le calcul de sensibilité est fait dans le modèle continu. Une fois discrétisé, le gradient de J $(\nabla J)_h$ n'est le gradient d'aucune fonctionnelle. En général, discrétisation et dérivation ne commutent pas, *ie* $(\nabla J)_h \neq \nabla_h J_h$. Il se peut que $(\nabla J)_h$ ne fournisse pas une direction de descente : peu importe le pas de descente choisi τ, $J(\Omega_{\tau\boldsymbol{\theta}}) > J(\Omega)$. Ce phénomène est exposé dans [108] ou encore [40], où même avec un petit nombre de paramètres, le gradient discret de J obtenu par différences finies

$$\frac{J(x + \delta x_i) - J(x)}{\delta x_i}$$

diffère du gradient continu discrétisé. Une solution consiste à discrétiser plus finement lorsqu'on détecte qu'une direction n'est pas une direction de descente, mais cela nécessite un recalcul de la triangulation. C'est la solution que nous avons adoptée.

Croisements d'arêtes Il se peut que lors de la phase d'advection de maillage, deux arêtes s'intersectent, ce qui donne lieu à un maillage non-conforme et donc inutilisable pour des calculs d'éléments finis. A chaque advection, on vérifie que $\mathcal{T}^{n+1}$ est bien conforme. Si ce n'est pas le cas, on recalcul une triangulation. Afin d'éviter que ce cas ne se produise trop souvent, un terme d'élasticité linéaire $Ae(\mathbf{u}) : e(\mathbf{v})$ (loi de Hooke) apparaît dans le produit scalaire (2.31). Ce terme fait apparaître la divergence du champ de déplacement, ce qui pénalise la compression des mailles, évitant des distorsions locales trop importantes.

Les essais numériques nous ont permis de constater que gradient $L^2(\Gamma)$ donné par le Théorème 2 était discontinu au voisinage d'un coin entre le bord Γ et le bord Γ_{out}. Nous avons ajouté un terme faisant intervenir le produit scalaire H^1 sur Γ pour régulariser le champ de déplacement dans ces zones.

Dépendance de la solution par rapport au niveau de

raffinement Il est crucial de vérifier que le domaine obtenu à l'issue de l'algorithme dépend peu de la finesse du maillage h. Pour ce faire, nous avons initialisé Ω^0 avec des maillages de finesses différentes puis comparé les résultats, à savoir Ω^* et $J(\Omega^*)$ afin de vérifier que lorsque $h \to 0$, les solutions discrètes convergent (Section 2.5.7).

1.1.5 Avantages et limitations de l'algorithme

Par construction, l'algorithme que nous proposons ne permet pas de changement de topologie. Par exemple, la forme finale ne peut pas faire apparaître de trous si la forme initiale n'en contient pas. En cela, l'espace des designs est limité par la topologie du domaine initial "initial guess". Néanmoins, nous considérons que pour les équations de Stokes ou Navier-Stokes les phénomènes d'homogénéisation ne se produisent pas lorsqu'on choisit une suite minimisante pour une fonctionnelle "raisonnable". Ce n'est pas le cas en mécanique des structures, voir [10] à ce sujet. Par conséquent, il est raisonnable de penser que le domaine initial a la bonne topologie.

L'algorithme que nous présentons a l'avantage d'être relativement simple, en particulier l'implémentation tient en moins de 400 lignes environ. Contrairement à une approche "Discretize then differentiate", il ne nécessite pas de connaître la méthode de discrétisation employée par le solveur du modèle fluide. En conséquence, à l'implémentation, il n'est pas nécessaire de connaître le solveur utilisé. En contrepartie, il faut faire à la main le calcul de la dérivée de forme, ce qui peut parfois être technique.

1.1.6 Exemple : énergie minimisée par une structure ramifiée

A titre d'illustration, nous présentons les résultats obtenus pour la minimisation de l'énergie dissipée dans une structure ramifiée.

On impose un profil parabolique sur chaque composante de Γ_{in} (Figure 1.4). La fonctionnelle commence par décroître fortement

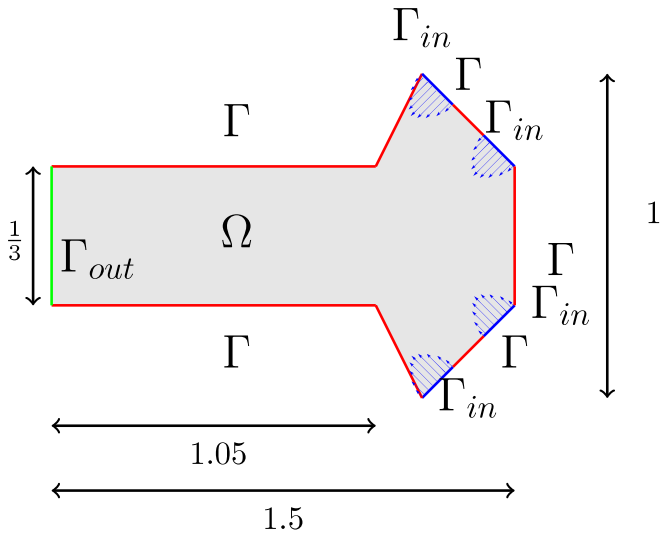


FIGURE 1.4 – *Schéma du cas-test "structure ramifiée", avec 4 entrées* Γ_{in} *et une sortie* Γ_{out}

(itération 80), mais la contrainte de volume n'est pas respectée. Le multiplicateur de Lagrange et le paramètre b augmentent, augmentant l'importance de la contrainte dans $\mathscr{L}$. Jusqu'à l'itération 400, le volume décroît en jusqu'à être en deçà de la valeur cible, puis augmente à nouveau légèrement pour permettre une nouvelle diminution du coût, et ainsi de suite jusqu'à convergence. A convergence, la fonction coût a diminué d'environ 20% (Figure 1.6).

1.2 Modélisation et optimisation géométrique d'une aquaporine

1.2.1 Présentation du problème

Les aquaporines sont des protéines des membranes présentes dans les cellules des plantes, des bactéries et des animaux, y-compris l'homme. Au sein de la bi-couche lipidique, cette protéine permet la filtration de des ions et d'autres solutés ainsi que la diffusion membranaire à travers la paroi cellulaire (voir Fig. 1.7).

Dans ce chapitre, nous nous intéressons à des problèmes d'optimisation de la forme d'une aquaporine. Ces problèmes sont similaires

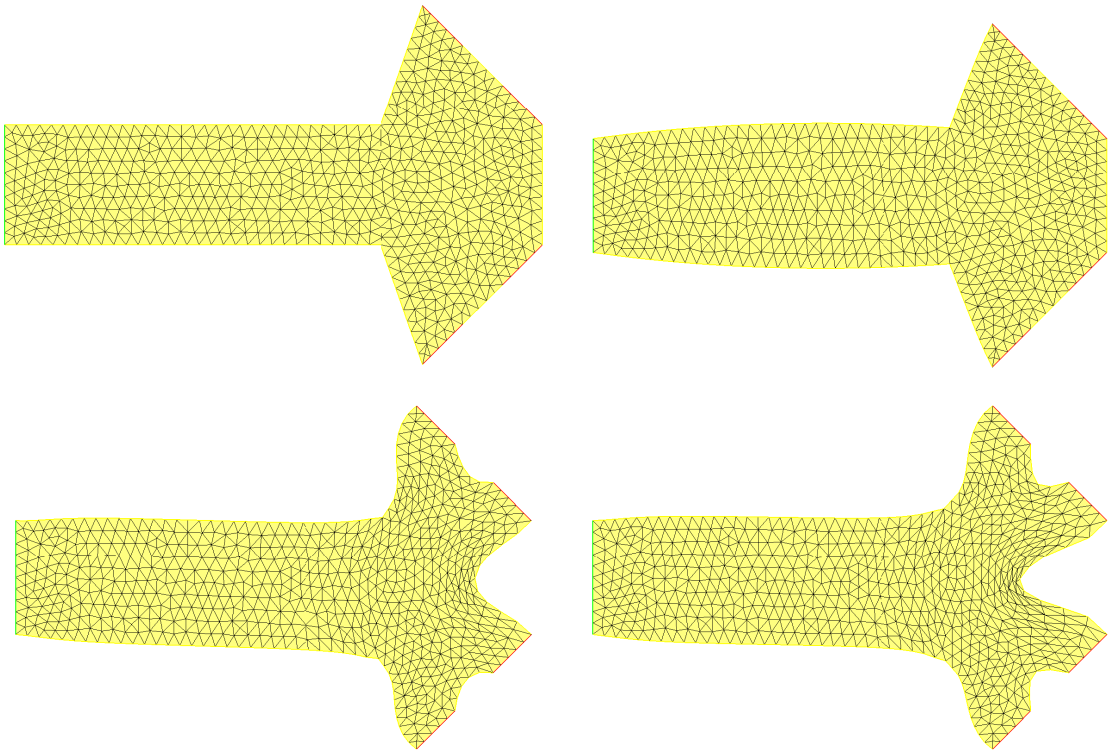


FIGURE 1.5 – *Formes successives Ω^n aux itérations $n = 0, 5,\ 240,\ 1000$ pour le cas de l'optimisation de la forme d'une structure ramifiée pour le critère "Energie dissipée" Section 2.5.2.*

à ceux abordés au Chapitre 2, avec quelques différences notables. Premièrement, le modèle fait apparaître des conditions aux limites de type "glissement partiel" avec une condition de non-pénétration.

$$\begin{aligned} [\sigma(\mathbf{u}, p)\mathbf{n} + \beta\mathbf{u}]_\tau &= 0 \\ \mathbf{u} \cdot \mathbf{n} &= 0 \end{aligned} \tag{1.10}$$

Deuxièmement, le modèle fait apparaître une condition de débit :

$$\int_{\Gamma_{in}} \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 = Q \tag{1.11}$$

Deuxièmement, la condition (1.11) ne ferme pas le système de Stokes, nous avons donc trouvé dans la littérature une façon naturelle d'imposer cette contrainte de façon à avoir un système bien posé, voir Sous-section 3.2.1.

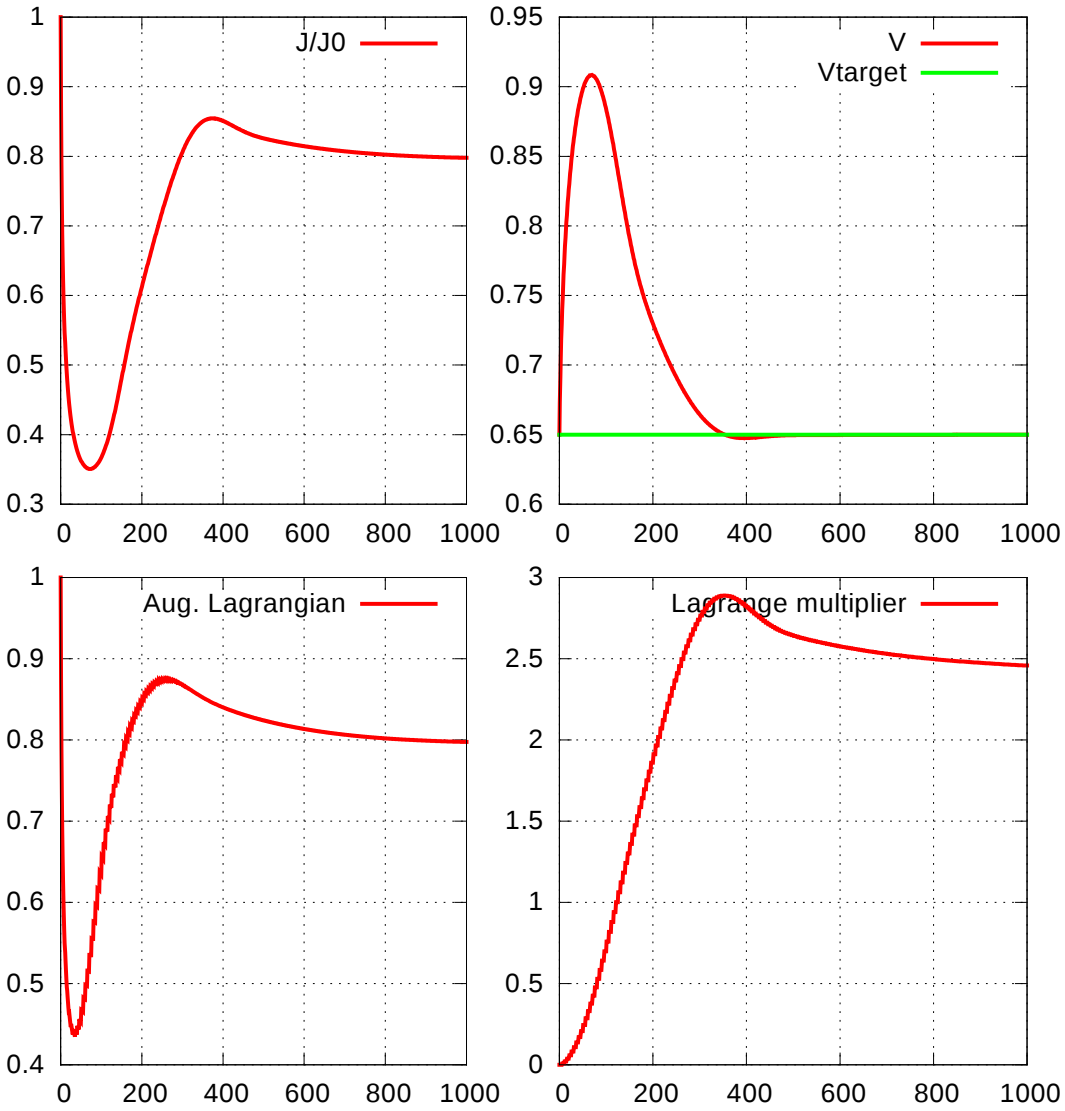


FIGURE 1.6 – *Graphiques de convergence (de gauche à droite, de haut en bas)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *et* ℓ^n *dans le cas de l'optimisation d'une structure ramifiée le critère "Energie dissipée" Section 2.5.2.*

Après quelques manipulations, on obtient le système modèle (1.12).

$$\begin{cases} -\operatorname{div}(2\nu\, e(\mathbf{u}_\lambda)) + \nabla p_\lambda &= 0 & x \in \Omega, \\ \operatorname{div}(\mathbf{u}_\lambda) &= 0 & x \in \Omega, \\ \mathbf{u}_\lambda &= 0 & x \in \Gamma_0, \\ \sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} + \lambda \mathbf{n} &= 0 & x \in \Gamma_{in}, \\ \sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} &= 0 & x \in \Gamma_{out}, \\ [\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} + \beta \mathbf{u}_\lambda]_\tau = 0 \quad , \quad \mathbf{u}_\lambda \cdot \mathbf{n} = 0 & & x \in \Gamma_1, \\ [\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}]_\tau = 0 \quad , \quad \mathbf{u}_\lambda \cdot \mathbf{n} = 0 & & x \in \Gamma_2. \end{cases} \tag{1.12}$$

avec λ choisi tel que la condition de débit (1.11) soit satisfaite. Une fois ce système précisé, on s'intéresse au problème d'optimisation par rapport au domaine du critère donné par :

$$J(\Omega) = 2\nu \int_\Omega |e(\mathbf{u}_{\Omega,\lambda})|^2 \, \mathrm{d}x + \beta \int_{\Gamma_1} |\mathbf{u}_{\Omega,\lambda}|^2 \, \mathrm{d}\mathcal{H}^1, \tag{1.13}$$

où $\mathbf{u}_{\Omega,\lambda}$ est solution du problème (1.12). Le choix de ce critère repose sur des considérations physiques : il représente l'énergie dissipée au sein du fluide dans Ω et par frottement au contact de la surface Γ_1. D'un point de vue mathématique, le critère étudié est l'énergie naturelle du système (1.12).

1.2.2 Objectifs du Chapitre 3

L'objectif de ce chapitre est triple. Il s'agit d'une part de préciser la modélisation mathématique d'un problème d'écoulement au travers d'une aquaporine. En effet, la modélisation mathématique du problème s'appuie sur les équations de Stokes avec des conditions de bord particulières : glissement partiel sur la partie du bord à optimiser et condition de débit sur une partie fixe du bord. Il s'agit de préciser cette formulation dans un cadre mathématique précis, notamment sur les conditions de bord, puis de montrer que le modèle obtenu est bien posé.

Dans un second temps, nous étudions le problème d'optimisation de forme qui consiste à minimiser l'énergie dissipée. Deux axes d'étude sont proposés. Le premier est de nature théorique : il s'agit de montrer que le problème d'optimisation de forme est bien posé

dans un cadre à définir. Le second est d'ordre numérique : il consiste à élaborer et à mettre en œuvre des méthodes numériques d'optimisation de forme. Ce travail s'inscrit dans le prolongement de l'article [73] dans lequel la forme de l'aquaporine est optimisée seulement par rapport à l'angle ou [19] dans lequel trois paramètres de design sont utilisés. L'idée est d'élargir l'espace des designs pour faire moins de suppositions *a priori* sur la forme de l'aquaporine.

1.2.3 Motivations applicatives

La première aquaporine a été mise en évidence en 1992 [23], elle a été nommée AQP1. Depuis, au moins dix autres types d'aquaporines ont été découverts chez les mammifères, dans le foie, les yeux, et les vaisseaux sanguins [133]. Des expériences ont montré que chez l'homme et la souris, AQP1 est impliquée dans de nombreux processus biologiques tels que la concentration de l'urine, le maintien de la pression crânienne et la production d'un fluide aqueux dans l'œil [41].

Selon certains auteurs, des médicaments agissant sur les aquaporines pourraient constituer des traitements contre les œdèmes, le cancer, l'épilepsie et les glaucomes [135]. Depuis leur découverte, les aquaporines sont le sujet d'une recherche abondante autant pour décrire leur structure que pour comprendre leur fonction. Pour une revue complète sur ce sujet, nous renvoyons à [41].

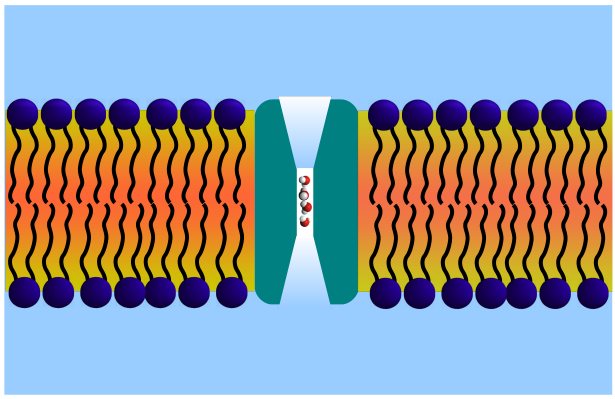


FIGURE 1.7 – *Schéma d'une aquaporine. Les molécules d'eau sont transportées dans un canal à travers la membrane cellulaire, par une aquapore en forme de sablier. Courtoisie de François Detcheverry (ILM, Lyon 1).*

D'un point de vue technologique, l'investigation de systèmes artificiels performants énergétiquement mène à de nombreuses applications telles que la filtration par une membrane. Les applications industrielles comprennent le dessalement de l'eau, l'industrie agro-alimentaire ainsi que le traitement des eaux usées.

Lorsqu'une membrane est utilisée, de l'eau salée ou saumâtre est injectée à haute pression dans une membrane semi-perméable qui filtre les composants non désirés. L'osmose inverse est couramment utilisée pour le dessalement [67].

Plus généralement, plusieurs technologies fondées sur le même principe permettent l'élimination de particules solides à différentes échelles de grandeur : la micro-filtration (0.1 µm–1 µm), la nano-filtration (3 nm–30 nm) et l'osmose inverse (0.1 nm–2 nm) [100]. La nano-filtration est un procédé fiable pour éliminer la plupart des pesticides, micro-polluants, virus et bactéries d'eaux souterraines ou de surface en vue de fournir de l'eau potable [55].

1.2.4 Méthode numérique d'optimisation de la forme d'une aquaporine

Nous avons utilisé des paramètres physiques similaires à ceux utilisés dans [19, 73], ce qui permet une comparaison des résultats. Néanmoins, la comparaison ne peut être que quantitative puisque

les articles cités fondent leurs résultats sur une simulation axisymétrique, et que nous obtenons des résultats à partir d'une géométrie plane. L'optimisation est menée en deux étapes. Dans un premier temps, l'angle des cônes est optimisé comme dans [73].Dans un second temps, en partant de l'angle trouvé à l'étape précédente, la forme des cônes est optimisée en utilisant des calculs de dérivée de forme pour la fonctionnelle d'énergie. L'expression de la dérivée de forme (3.4.3) est quelque peu inhabituelle et nécessite d'introduire un traitement spécifique qui diffère par exemple du Chapitre 2.

Afin d'obtenir des formes réalistes, nous imposons des conditions de symétrie axiale puis axisymétrique sur la forme de l'aquaporine. Ce chapitre a pour objet de d'étendre les travaux [19]–[73] en agrandissant l'espace des formes admissibles. Il est important de remarquer que ces travaux étudient la résistance hydrodynamique tandis que ce travail porte sur l'étude de l'énergie dissipée. Ces deux quantités sont liées, mais nous avons considéré l'énergie dissipée qui est l'énergie naturelle du système (1.12).

Les limitations de notre choix de modélisation résident dans le fait que les contraintes de manufacture à l'échelle de quelques atomes ne sont pas prises en compte. Une autre limitation concerne les particules solides, dont on sait qu'elles influence le comportement du fluide lorsque leur fraction volumique devient significative [92, Chapter 10]. Ces particules solides ne sont pas prises en compte dans la modélisation.

Nous traitons à la fin du chapitre le cas axisymétrique, qui est plus pertinent d'un point de vue physique, puisqu'il revient à considérer non plus une aquapore infinie selon l'axe perpendiculaire au plan, mais bien une aquapore cylindrique. Ce choix introduit quelques difficultés supplémentaires détaillées dans la partie 3.6.

Chapter 2

Geometry optimization of a Navier-Stokes flow

This chapter is based on the paper "Geometrical shape optimization in fluid mechanics using freefem++", C. Dapogny, P. Frey, F. Omnès, and Y. Privat, to appear in Structural and Multidisciplary Optimization, 2017. [50]

This section is organized as follows. In Section 2.2, we introduce the 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 2.4, we describe in more details the main ingredients of the proposed numerical method: after a short motivating outline in Section 2.4.1, we discuss the salient features of our shape optimization algorithm in Sections 2.4.2, 2.4.3, 2.4.4, 2.4.5 and 2.4.6; a sketch of this algorithm is then provided in Section 2.4.7. Section B is then a short guide of our practical implementation; it is expected that, together with the thorough comments left throughout our code, this will allow the user to define and solve his own shape optimization test cases in a user-friendly way. In Section 2.5, we introduce and comment five test cases which are dealt with by our algorithm. Finally, Section 2.6 concludes by evoking limitations of our approach as well as perspectives for possible improvements and extensions.

2.1 State of the art

2.1.1 Geometry optimization

Shape optimization problems arise naturally in many fields such as structural mechanics, electromagnetism and fluid mechanics. In each context, one aims at maximizing or minimizing a criterion with respect to geometrical parameters, or even to the whole shape of the domain of interest. Historically, the first applications came with the development of aeronautics. Let us briefly outline the main features of the most popular shape optimization strategies in the literature, without claiming for exhaustivity. For more in-depth discussions in the context of fluid mechanics, we refer to [77, 108], or to the review paper [107].

Developments in aeronautics 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 [42]; see also [86, 87], and [122] where optimal profiles for minimum drag problems are calculated thanks to shape sensitivity analyses. We generally refer to [77, 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 [47] about the numerical optimization of a cooling fan - or in computational biology: for instance, in [3, 5], the design optimization of an artery graft for preventing the formation of a stenosis is investigated from a numerical point of view.

More recently, shape optimization problems in fluid mechanics have emerged from bio-medical engineering. Recent works present the optimization of bronchial trees using reduced models [106, 52]. Optimizing the shape of a full bronchial tree without using reduced models is still a challenge due to the high computational costs,

notably caused by a complex geometry featuring different length-scales.

In several contexts, in particular when the criterion to optimize appears too difficult to handle, "derivative-free" approaches are preferred [105, 101, 126]. In such works, global optimization methods such as genetic algorithms are often considered. Rarely, models can be reduced to one single parameter, in which case the optimization is made by locating the minimum on a function graph [45]. More sophisticated variants make use of an empirical method [129]. However, in most studies, the algorithms rely on a notion of differentiation. Therefore, shape sensitivity analysis is at the heart of shape optimization. In such an analysis, one deals with the variation of the criterion under a slight modification of a geometrical parameter.

2.1.2 Problem discretization

Two standard approaches compete in shape sensitivity analysis: Discretize then Optimize and Optimize then Discretize.

Discretize then Optimize In *Discretize then Optimize* (DTO) approaches, the sensitivity analysis of the quantities of interest (drag, lift, and so on) is performed on a discretized fluid model. For instance, the boundary of the domain is in general parametrized using curves or lines, the parameters are the geometrical coordinates of the vertices or control nodes. This method is conceptually simple and, in general, falls into finite dimensional optimization theory. This drives to use a wide range of numerical methods (e.g conjugate gradient, BFGS, simplex method) [107, 98].

Because of its mathematical simplicity, DTO is often used in engineering [114, 134, 136]. An example of result from [136] is shown in Figure 2.1

Optimize then discretize In *Optimize then Discretize* (OTD) approaches, a shape sensitivity analysis is performed on a continuous model and cost functional. This framework is more mathematical and is coupled with advanced tools from the calculus of variations. In spite of their efficiency, the resulting algorithms remain

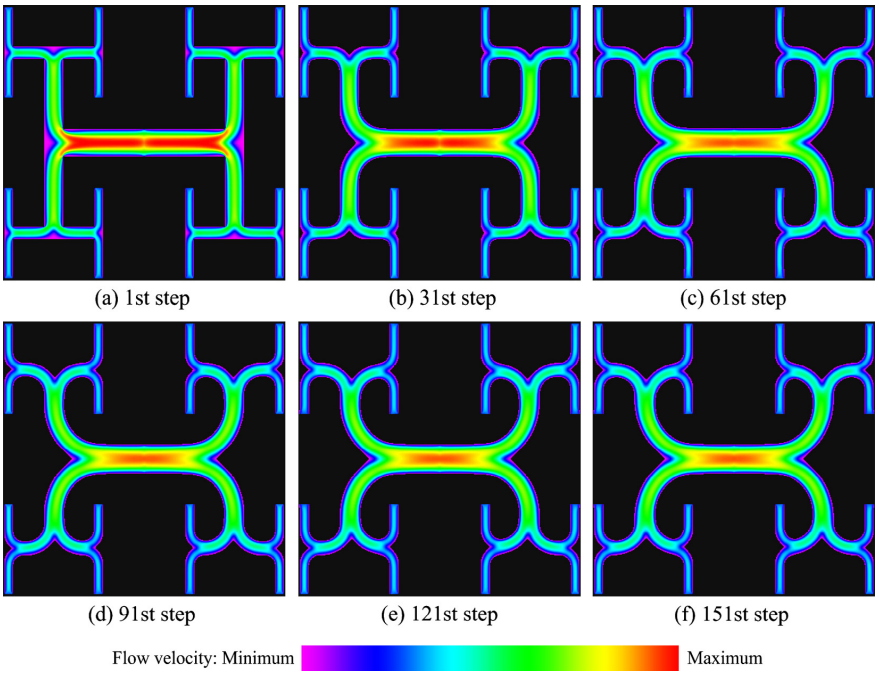


Figure 2.1 – *Example of shape optimization of a fluid distributor. The flow is modelled using turbulent Lattice-Boltzmann equations at Re 100-30000. Fluid and solid cells at the boundary are iteratively swapped using a heuristic criterion until a residual falls under a fixed threshold. Credits : L. Wang, Y. Fan and Lingai Luo [136]*

in general conceptually simple and are inspired by gradient-like descent methods. Advantages of OTD include more flexibility in the discretization in the following sense: changing the discretization parameters does not influence the sensitivity analysis of the problem. Furthermore, no *a priori* assumption is made on the optimal shape, which allows to explore a wider range of shapes. The level-set method [117, 38] allows for some topology changes including "coalescence", a phenomenon in which two parts of the boundary join and merge. However, the creation of holes within the domain is not naturally handled : this is known as the "nucleation problem". Two methods based on the topological gradient have been devised to avoid this drawback. In the first method, the topological gradient is used to determine where holes should be created, the holes are kept if they allow a decrease in the cost-function value, they are otherwise removed [11]. In the second method, the topological gradient is added into the Hamilton-Jacobi equation that controls the level-set motion, see [138, 39]. OTD has been much used in academic publications [16, 33, 59, 13]. More applied studies include [61] and [99]. See Figure 2.2 extracted from [99].

Within the scope of OTD, [131] provides a detailed reference for sensitivity analysis, introducing the main topics of Eulerian and Material derivatives, necessary for the sensitivity analysis of PDEs. Results and techniques on existence and regularity of optimal shapes, as well as sensitivity analysis can be found in [81] (in French). Lastly, the book [8] (in French) provides an introduction to the numerical aspects of shape optimization. This is the approach we choose in this work.

Discussion In this section, we briefly discuss the particularities of each approach. Reference [118] provides a survey on sensitivity analysis, including a full discussion on the OTD and DTO for the Euler equations, but most considerations remain valid for the incompressible Navier-Stokes equations as well.

Unlike OTD, DTO does not feature *inconsistent gradients*. When discretizing the (continuous) shape gradient and shape functional,

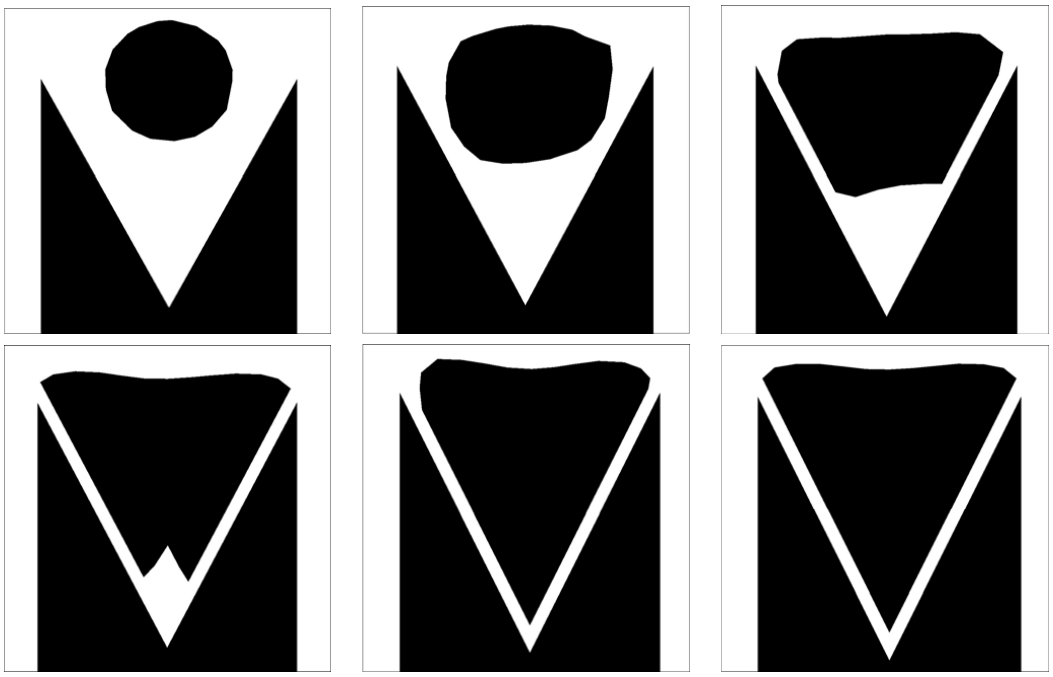


Figure 2.2 – *In this example, the level-set method is used to maximize the ferromagnetic repulsive force between a fixed object (bottom) and a deformable ferromagnetic material (top). Credits : H. W. Kuhn and A. W. Tucker, [99].*

discretization errors can appear such the gradient is inconsistent. Paper [40] illustrates this point quite well for the non-linear Navier-Stokes equations : with only two design parameters, the functional isolines and gradients are shown not to be orthogonal. Using the gradient of the discretized functional (for example using automatic differentiation), the computed gradient is exact, so no inconsistencies can appear.

DTO comes with a practical difficulty of implementation : one needs a precise knowledge of the numerical solver for PDEs. For example, if one uses Finite Elements, it is necessary to know which quadrature rule is used and the matrix assembly process. On the other hand, the mathematical concepts behind the sensitivity analysis are more elementary : only partial derivatives with respect to real variables are considered. For ODT, one must compute shape-derivatives or sensitivity analysis with respect to continuous parameters, which comes with technical difficulties.

2.1.3 Sensitivity analysis : adjoint methods

Any shape optimization method relies on a parametrization of shapes, that is, on the definition of a set of design variables. Depending on the situation, these design variables may be physical parameters of shapes (the length of some pipe, or thickness of a region), control points of a CAD description, or the vertices of a meshed representation.

In all cases, the sensitivity of the objective and constraint functionals of the optimization problem with respect to the design variables can be evaluated either by approximate methods. For instance, it may be computed by finite differences featuring small perturbations of the parameters. It may also be computed analytically, by relying on adjoint techniques from optimal control theory [90, 102, 123]. See also [70] for a review of adjoint methods in computational fluid dynamics. In this work, we choose to adopt this set of techniques.

2.1.4 Topology optimization

Recently, several strategies have been devised to circumvent the difficulty associated with mesh self-intersection, 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 [22] and references therein) -have been introduced in the context of fluid mechanics in [30]; see also [119] 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, 'gray-scale' values in $(0, 1)$. The flow equations have then to be given an appropriate meaning to

Figure 2.3 – *Evolution of a solid obstacle immersed in a cubic domain of Stokes fluid during the topology optimization procedure. A relaxation method is used to minimize the energy dissipated by the fluid. Credits : T. Borrvall and J. Petersson. [30]*

account for the presence of 'void' and 'gray-scale' regions. This is typically achieved by adding a ρ-dependent damping term (or Brinkman's law) to the flow equations [30], a heuristic inspired from the theory of porous media whereby the void is filled with a fluid with very low permeability, thus mimicking no slip boundary conditions at the interface between the fluid and void domains (Figure 2.3). See [76] and [63], then [69] for a generalization to the case of Navier-Stokes flows.

Let us eventually mention the contribution [96] where topology optimization problems are tackled in the context of the unsteady Navier-Stokes equations, and reveal the limitations of this penalization approach as far as the accuracy of the resolution of the flow equations is concerned.

Another class of shape and topology optimization strategies relies on the level set method, pioneered in [115], then introduced in structural optimization in [128, 12, 137]. Such methods describe a shape Ω via the use of a scalar function ϕ defined on the whole computational domain D: the negative sub-domain of ϕ coincides with Ω, while its positive sub-domain accounts for void (or, in

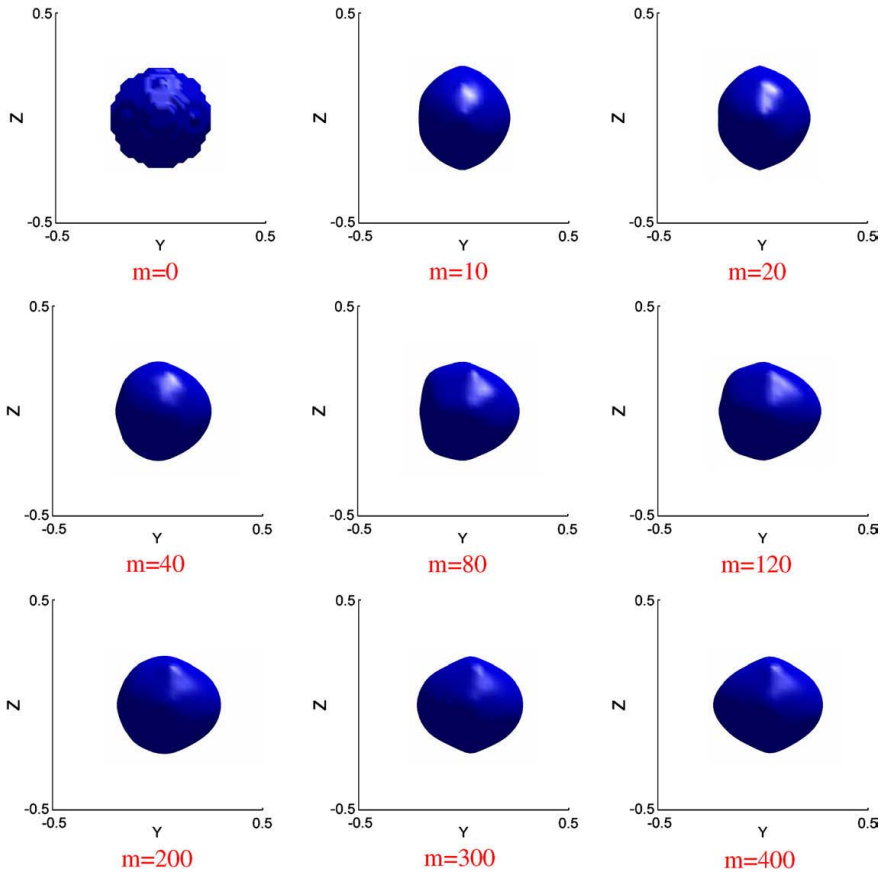


Figure 2.4 – *Evolution of a solid obstacle immersed in a cubic domain of fluid during the minimization procedure. A 3D level-set method is used to minimize the energy dissipated by the fluid. Credits : S. Zhou and Q. Li [139]*

practice, another fluid with low permeability, according to the aforementioned 'Brinkman' penalization approximation). In the two-dimensional work [60], 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 [139] in the three-dimensional setting, see Figure 2.4.

See also [34] for another use of the Level Set method in the context of Navier-Stokes flows. Recent contributions have proposed alternative efficient level set methods where the flow equations are solved by the Lattice Boltzmann method [119, 95] or the Extended Finite Element method [94], alleviating the need for the 'Brinkman'

penalization method. On a different note, in [43], 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 level set methods, except for the fact that they bring into play shapes, or phases, with 'thickened boundaries' [68]. See figure 2.5 for an illustration of the results obtained from the phase-field method.

2.1.5 Homogenization methods

Homogenization methods for shape optimization are based on a homogenized model, in which microscopic properties are included into a macroscopic equation. Homogenization theory was first investigated for structural mechanics, and it was proposed to model the mechanical properties of bones [15].

In the seminal works of Bendsøe, Kikuchi, Suzuki *et al.* in structure mechanics [132, 21] the approach is to allocate a microscopic property within the domain. Since this property appears in the macroscopic equations in the form of elasticity coefficients, the goal is to optimize these coefficients. The book [10] discusses numerical algorithms and remedies to numerical issues in great detail again for structure mechanics.

The Stokes and Navier-Stokes equations have also been investigated in the scope of homogenization in [7] with slip boundary conditions. For the Stokes equations, depending on the asymptotic size of the holes as the size parameters tends to 0, the limit model is proven to be Stokes, Brinkman or Darcy. The convergence is strong (resp. weak, strong) in the appropriate spaces. In all cases, the limit model is fully explicit. In the critical size case, the Navier-Stokes solution is proven to converge weakly to the solution a Brinkman model.

In [6], similar conclusions are reached with Dirichlet boundary

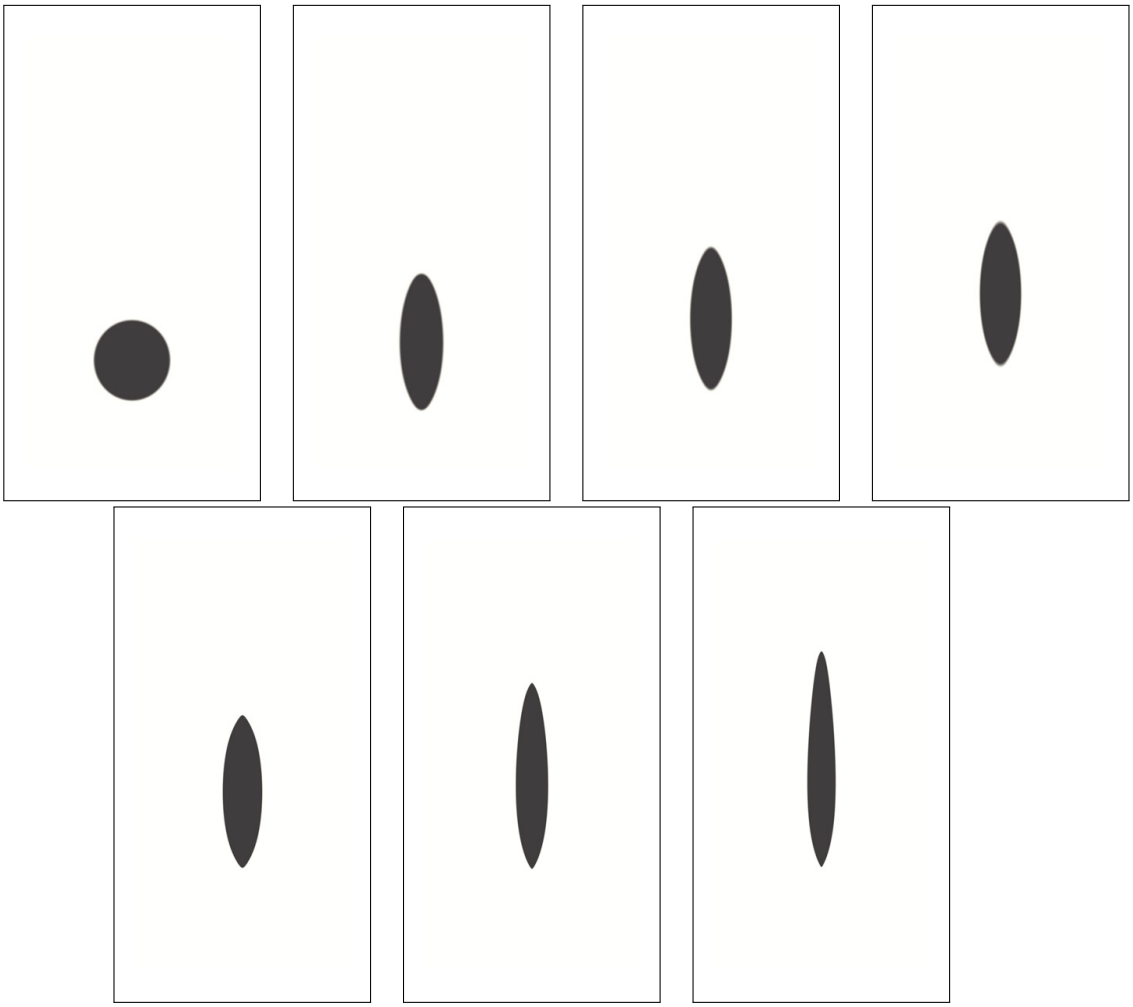


Figure 2.5 – *Evolution of a solid obstacle immersed in a rectangular domain of fluid during the minimization procedure. A phase-field method is used to minimize the energy dissipated by the fluid. Credits : H. Garcke, C. Hecht, M. Hinze and C. Kahle [68]*

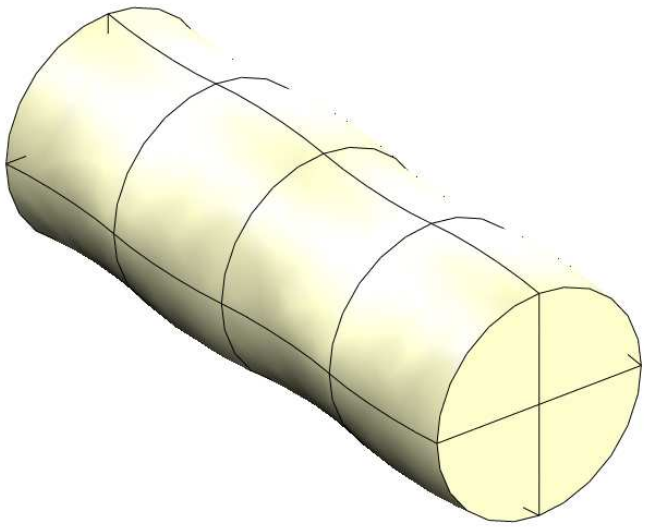


Figure 2.6 – *3D shape with a lower dissipated energy than the cylinder. Credits : A. Henrot and Y. Privat [83]*

conditions.

2.1.6 Theoretical issues

From a theoretical point of view, the existence of solutions to the geometry optimization problem of dissipated energy for incompressible Navier-Stokes equations has only been proven under uniform regularity assumptions on admissible domains [78].

Qualitative information about minimizers can sometimes be found. For example, in [83], it is proven the minimizer to the dissipated energy minimization problem is not a cylinder. The proof relies on optimality conditions involving the shape derivative of the criterion. An example of domain associated to a lower dissipated energy was numerically computed 2.6.

2.2 Statement of the shape optimization problem

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 chapter arise in two space dimensions (see Section 2.5), 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 giving up simplicity and clarity.

Let $(\mathbf{u}, p)$ is the solution to the stationary Navier-Stokes system.

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

This equation describes the behavior of an Newtonian fluid, at low Reynolds. Note that the kinematic viscosity is related to the Reynolds number *via* $\nu = \frac{1}{Re}$. In the numerical applications, only laminar flows are investigated, therefore we choose $Re < 1000$.

The shape optimization problem of interest reads

$$\min_{\Omega\in\mathcal{O}_{\text{ad}}} J(\Omega) \text{ s.t. } G(\Omega) = 0. \tag{2.2}$$

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.

$$E(\Omega) = \int_\Omega \sigma(\mathbf{u}, p) : e(\mathbf{u})\, dx = 2\nu \int_\Omega \|e(\mathbf{u})\|^2\, dx, \tag{2.3}$$

- A least-square discrepancy on the outlet boundary Γ_{out} between the velocity $\mathbf{u}$ of the fluid, solution to the Navier-Stokes system (2.1), and a given reference profile $\mathbf{u}_{\text{ref}}$.

$$D(\Omega) = \frac{1}{2}\int_{\Gamma_{\text{out}}} |\mathbf{u} - \mathbf{u}_{\text{ref}}|^2\, ds \tag{2.4}$$

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 [17, 103].

- A least-square vorticity functional on Ω quantifying the total vorticity of the flow.

$$C(\Omega) = \int_\Omega \omega(\mathbf{u})^2 \, dx \tag{2.5}$$

where $\omega(\mathbf{u}) = \nabla \times \mathbf{u}$. If $C(\Omega) = 0$, the flow is irrotational.

Remark 1. *Functionals C and E are related. In [97], the following equality is given for incompressible flows governed by the Navier-Stokes equations*

$$2\nu \int_\Omega \|e(\mathbf{u})\|^2 \, dx = \nu \int_\Omega \omega(\mathbf{u})^2 \, dx - 2\nu \int_{\partial\Omega} (\mathbf{u} \times \omega(\mathbf{u}) - \nabla \frac{u^2}{2}) \cdot \mathbf{n} \, ds \tag{2.6}$$

If boundary effects are neglected, the energy dissipation is due to the vorticity. A precise explanation on this relationship and a proof are given in [93].

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 (2.2) reads:

$$\mathcal{O}_{ad} = \{\Omega \subset \mathbb{R}^d, \text{ open, regular and bounded, such that } \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \subset \partial\Omega\} \tag{2.7}$$

Last but not least, as far as the constraint functional $G(\Omega)$ is concerned, we shall restrict ourselves to equality constraints on the volume $\text{Vol}(\Omega) = \int_\Omega dx$ or the perimeter $\text{Per}(\Omega) = \int_{\partial\Omega} ds$ 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 global minimizers of problems of the form (2.2) 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 or adding constraints on the perimeter. One way to prove the existence of optimal shapes for energy dissipation with the Navier-Stokes system is to impose uniform Lipschitz regularity on the domain, see [78]. On the contrary, one may enlarge the set of admissible domains, so that it includes 'density functions', and not only 'black and white' shapes. See for instance [35, 81, 131] about these issues, or [83, 24] in the context of fluid mechanics.*

In numerical practice, one is often interested in searching for local minimizers of (2.2), which are close to an initial guess inspired by physical intuition or expertise gained from experiments. These are the 'optimal' shapes which are typically delivered by local optimization methods, such as the steepest-descent algorithms used in the present chapter.

2.3 Shape sensitivity analysis using Hadamard's boundary variation method

Most optimization algorithms - such as steepest-descent methods - 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 (2.2). 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).

The approach described in this chapter belongs to the second category, and therefore requires to compute derivatives with respect to the domain.Several ways exist to define a notion of shape derivative, and we rely on Hadamard's boundary variation method, a brief sketch of which is now provided; see for instance to [81], Chap. 5, or [8, 110] for in-depth expositions. See also [112] for an overview of the notion of *topological derivative*, and [14] 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 assessed with respect to small perturbations of its boundary: variations of a given shape Ω are considered in the form

$$\Omega_{\boldsymbol{\theta}} = (\mathrm{Id} + \boldsymbol{\theta})(\Omega), \tag{2.8}$$

where $\boldsymbol{\theta} : \mathbb{R}^d \to \mathbb{R}^d$ is a 'small' vector field, and Id is the identity mapping from $\mathbb{R}^d$ into itself; see Figure 2.7 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:

$$\Theta_{ad} = \left\{ \boldsymbol{\theta} : \mathbb{R}^d \to \mathbb{R}^d \text{ smooth} \right\};$$

so that variations (2.8) of admissible shapes stay admissible.

Remark 3. *In practice, we also impose that $\boldsymbol{\theta} = 0$ on $\Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ so that only Γ is allowed to move.*

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*

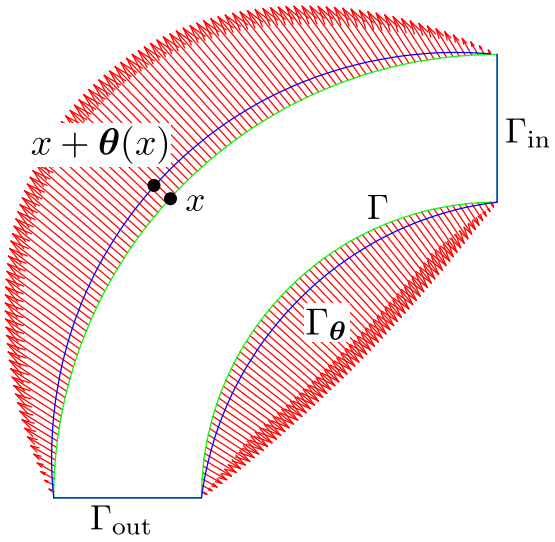


Figure 2.7 – *Example of a variation $\Omega_{\boldsymbol{\theta}}$ of a shape Ω. First iteration of the bending pipe example, Section 2.5.1. The initial domain is represented in blue, the deformed one is in green. Note : the arrows are not to scale.*

expansion holds:

$$F(\Omega_{\boldsymbol{\theta}}) = F(\Omega) + F'(\Omega)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}) \tag{2.9}$$

where $o(\boldsymbol{\theta}) \to 0$ as $\boldsymbol{\theta} \to 0$.

When it comes to shape derivatives, the first result of interest deals with the volume and perimeter functionals; see [8, 81] 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 (2.3) or (2.4) 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 [8] for a comprehensive introduction to such techniques in the context of shape optimization.

Theorem 2. *Let $\Omega \in \mathcal{O}_{ad}$; then,*

(i) The energy dissipation $E(\Omega)$ given by (2.3) is shape differentiable and its derivative reads:

$$\forall \boldsymbol{\theta} \in \Theta_{ad},\ \ E'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \left(-2\nu e(\mathbf{u}) : e(\mathbf{u}) + 2\nu e(\mathbf{u}) : e(\mathbf{v}_e)\right) \boldsymbol{\theta} \cdot \mathbf{n}\, ds, \quad (2.10)$$

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

$$\left\{ \begin{array}{rcll} -\nu\Delta \mathbf{v}_e + (\nabla \mathbf{u})^T \mathbf{v}_e - (\nabla \mathbf{v}_e)\mathbf{u} + \nabla q_e &=& -2\nu\Delta \mathbf{u} & x \in \Omega, \\ \mathrm{div}(\mathbf{v}_e) &=& 0 & x \in \Omega, \\ \mathbf{v}_e &=& 0 & x \in \Gamma \cup \Gamma_{\mathrm{in}}, \\ \sigma(\mathbf{v}_e, q_e)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_e &=& 4\nu e(\mathbf{u})\mathbf{n} & x \in \Gamma_{\mathrm{out}}. \end{array} \right. \quad (2.11)$$

(ii) The least-square functional $D(\Omega)$ defined by (2.4) is shape differentiable and its derivative reads:

$$\forall \boldsymbol{\theta} \in \Theta_{ad},\ \ D'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma 2\nu e(\mathbf{u}) : e(\mathbf{v}_d)\, \boldsymbol{\theta} \cdot \mathbf{n}\, ds, \quad (2.12)$$

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

$$\left\{ \begin{array}{rcll} -\nu\Delta \mathbf{v}_d + (\nabla \mathbf{u})^T \mathbf{v}_d - (\nabla \mathbf{v}_d)\mathbf{u} + \nabla q_d &=& 0 & x \in \Omega, \\ \mathrm{div}(\mathbf{v}_d) &=& 0 & x \in \Omega, \\ \mathbf{v}_d &=& 0 & x \in \Gamma \cup \Gamma_{in}, \\ \sigma(\mathbf{v}_d, q_d)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_d &=& \mathbf{u} - \mathbf{u}_{ref} & x \in \Gamma_{out}. \end{array} \right. \quad (2.13)$$

(iii) The vorticity functional $C(\Omega)$ given by (2.5) is shape differentiable and its derivative reads $\forall \boldsymbol{\theta} \in \Theta_{ad}$

$$C'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \left(\omega^2 - 2\omega(\mathbf{u})\frac{\partial \mathbf{u}}{\partial n} \times \mathbf{n} + 2\nu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}_c)\right)(\boldsymbol{\theta}\cdot\mathbf{n})\, ds \tag{2.14}$$

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

$$\begin{cases} -\nu\Delta\mathbf{v}_c + (\nabla\mathbf{u})^T\mathbf{v}_c - (\nabla\mathbf{v}_c)\mathbf{u} + \nabla q_c = \\ \qquad 2\begin{pmatrix}\frac{\partial(\nabla\times\mathbf{u})}{\partial x_2} & -\frac{\partial(\nabla\times\mathbf{u})}{\partial x_1}\end{pmatrix} & x \in \Omega \\ \qquad \operatorname{div}(\mathbf{v}_c) = 0 & x \in \Omega, \\ \qquad \mathbf{v}_c = 0 & x \in \Gamma \cup \Gamma_{\text{in}}, \\ \sigma(\mathbf{v}_c, q_c)\mathbf{n} + (\mathbf{u}\cdot\mathbf{n})\mathbf{v}_c = \\ \qquad 2(\nabla\times\mathbf{u})\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}\mathbf{n} & x \in \Gamma_{out} \end{cases} \tag{2.15}$$

Elements of proof of Theorem 2 are postponed to Appendix A for the sake of clarity. Notice that the calculation of $E'(\Omega)(\boldsymbol{\theta})$ can be found for instance in [83, Proposition 3.2]. Applying the same method allows to compute the shape derivative of the functional $D(\Omega)$.

Remark 4. *1. As is customary in shape optimization - and in optimal control in general -, the adjoint systems (2.11) and (2.12) are linear, while the original Navier-Stokes system (2.1) is non-linear. In particular, unlike the Navier-Stokes system, it can be solved at little computational cost using standard methods.*

2. From the mathematical point of view, the adjoint systems (2.11) and (2.13) are well-posed in suitable functional spaces when the parameter ν is assumed to be large enough (see e.g. [83]).

3. The use of adjoint method is widespread in shape and topology optimization [96, 125, 118]. This choice is motivated by computational cost. With the direct method, the sensitivity is computed

with respect to each of the N design parameters independently, requiring N systems to be solved. In the adjoint approach, only one similar system is solved for an equivalent individual cost, greatly reducing the number of system resolutions.

4. *Differentiability is proven in two steps. First, $J((I+t\boldsymbol{\theta})(\Omega))$ is rewritten on a fixed domain (independent of $\boldsymbol{\theta}$) using change of variables formulae. Then the implicit Banach-space version of the implicit function theorem is used to prove differentiability. See [89] for more details on the general principle, as well as an application to the stationary Navier-Stokes system.*

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:

$$F'(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \phi\, \boldsymbol{\theta}\cdot\mathbf{n}\, ds =: (\phi, \boldsymbol{\theta}\cdot\mathbf{n})_{L^2(\Gamma)}, \tag{2.16}$$

where the scalar function $\phi : \Gamma \to \mathbb{R}$ is the 'shape gradient' of Ff with respect to the $L^2(\Gamma)$ inner product. The shape derivative of smooth functionals can be written as in equation (2.16), according to the structure theorem, see [113]. This statement is referred to as the *Hadamard structure theorem* for shape derivatives; see [81, §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 (2.16) makes it easy to infer descent directions for $F(\Omega)$. Indeed, if $\boldsymbol{\theta}$ coincides with $-\phi\boldsymbol{n}$ on Γ, it readily follows from (2.9) that, for $t > 0$ small enough:

$$F(\Omega_{t\boldsymbol{\theta}}) = F(\Omega) - t\int_\Gamma \phi^2\, ds + o(t) < F(\Omega). \tag{2.17}$$

2.4 Numerical methods

In this section, we describe in more detail the numerical methods involved in the resolution of the shape optimization problem (2.2).

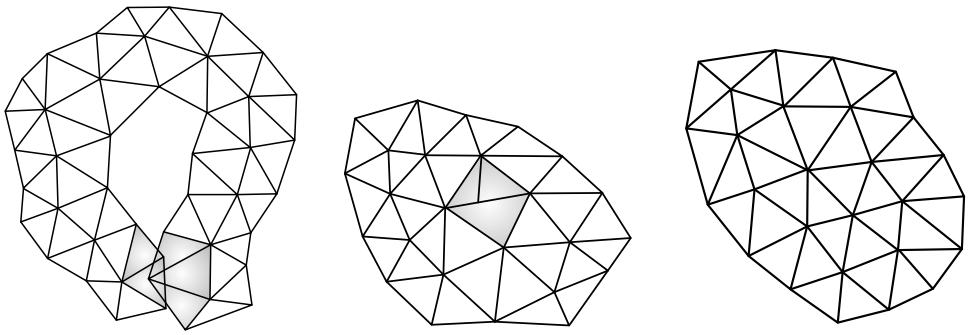


Figure 2.8 – *Examples of a mesh with overlapping triangles (in gray, left); a mesh with non overlapping, yet non conforming triangles (in gray, middle); a computational mesh, right.*

2.4.1 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. $\overline{\Omega} = \cup_{k=1}^{K} T_k$,
- 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 $\mathcal{T}$.

See Figure 2.8 for illustrations of these notions.

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 emphasize the needed numerical methods in the resolution of (2.2), we now give a deliberately hazy sketch of the main stages; a practical version is given in Section 2.4.7.

- **Initialization:** The initial domain Ω^0 is equipped with a mesh $\mathcal{T}^0$.
- **For $n = 0$, ... until convergence:**
  1. Compute the solution $(\mathbf{u}, p)$ of the Navier-Stokes equation (2.1), and the adjoint state $(\mathbf{v}, q)$, solution of (2.11) or (2.13) on Ω^n, using the mesh $\mathcal{T}^n$.
  2. Compute 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 (2.2).
  3. Choose a sufficiently small time step τ^n and update 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), (2.11) and (2.13) is by no means trivial; Section 2.4.2 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 2.4.3.
- 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 2.4.4 the stakes of mesh deformation, and in Section 2.4.5 a strategy for calculating a nice shape gradient which eases this purpose.

2.4.2 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 (2.11) and (2.13) are linear, so 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) in this section, which is in all regards more difficult.

Dealing with the nonlinear convective term using Newton's method

We rely on a fairly standard Newton method for nonlinear problems. Writing (2.1) in the abstract form

$$A(\mathbf{u}, p) = 0, \tag{2.18}$$

Newton's method achieves the solution as the limit of the sequence of linearized problem solutions $(\mathbf{u}^k, p^k)$, where each update $(\delta\mathbf{u}^k, \delta p^k)$ between the steps k and $(k+1)$ is calculated as the solution to the linearized version of (2.18) around $(\mathbf{u}^k, p^k)$:

$$d_{(\mathbf{u}^k, p^k)} A(\delta\mathbf{u}^k, \delta p^k) = -A(\mathbf{u}^k, p^k), \tag{2.19}$$

where $d_{(\mathbf{u}^k, p^k)} A$ is the linearization of 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 (2.19) reads as follows:

1. **Initialization:** The pair $(\mathbf{u}^0, p^0)$ is the solution to the Stokes counterpart of (2.1) (i.e. the version of (2.1) where the non linear term is omitted):

$$\begin{cases} -\nu\Delta\mathbf{u}^0 + \nabla p^0 = 0 & x \in \Omega, \\ \operatorname{div}(\mathbf{u}^0) = 0 & x \in \Omega, \\ \mathbf{u}^0 = \mathbf{u}_{\text{in}} & x \in \Gamma_{\text{in}}, \\ \mathbf{u}^0 = 0 & x \in \Gamma, \\ \sigma(\mathbf{u}^0, p^0)\mathbf{n} = 0 & x \in \Gamma_{\text{out}}. \end{cases} \tag{2.20}$$

2. **For** $k = 1, \ldots,$ $(\mathbf{u}^{k+1}, p^{k+1})$ 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 linear system (viz. (2.19)),

$$\begin{cases} -\nu\Delta(\delta\mathbf{u}^k) + (\mathbf{u}^k\cdot\nabla)(\delta\mathbf{u}^k) + ((\delta\mathbf{u}^k)\cdot\nabla)\mathbf{u}^k & \\ \quad +\nabla(\delta p^k) = \nu\Delta\mathbf{u}^k - (\mathbf{u}^k\cdot\nabla)\mathbf{u}^k - \nabla p^k & x\in\Omega, \\ \operatorname{div}(\delta\mathbf{u}^k) = 0 & x\in\Omega, \\ \delta\mathbf{u}^k = 0 & x\in\Gamma\cup\Gamma_{\text{in}}, \\ \sigma(\delta\mathbf{u}^k, \delta p^k)\mathbf{n} = 0 & x\in\Gamma_{\text{out}}, \end{cases} \tag{2.21}$$

3. **Ending criterion:** The algorithm ends when

$$e^k < \varepsilon_{\text{stop}}, \text{ with}$$

$$e^k := \sqrt{\frac{\|\delta\mathbf{u}^k\|^2_{L^2(\Omega)^d} + \|\nabla(\delta\mathbf{u}^k)\|^2_{L^2(\Omega)^{d\times d}}}{\|\mathbf{u}^k\|^2_{L^2(\Omega)^d} + \|\nabla\mathbf{u}^k\|^2_{L^2(\Omega)^{d\times d}}}} \tag{2.22}$$

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

This ending criterion is inspired from [71, Chapter 6], where the sequence $(\mathbf{u}^k, p^k)$ is proved to converge quadratically to the solution of (2.1), provided the initial pair $(\mathbf{u}^0, p^0)$ is 'close' enough to the latter. In other terms, the error e^k behaves as $e^{k+1} \approx \left(e^k\right)^2$. In practice, only 3 or 4 iterations are required to fulfill (2.22) with $\varepsilon_{\text{stop}} = 10^{-10}$.

Let us mention that other approaches have been devised 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 [71].

Remark 5. *Newton-like algorithms are well-known to experience difficulties as far as convergence is concerned, especially when the initial state is 'far' from the sought solution; in our 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 solution $(\mathbf{u}^0, p^0)$ *to the Stokes equation (2.20) is 'too far' from that* $(\mathbf{u}, p)$ *to (2.1). In such a case, one may resort to mixed strategies (e.g. with a Stokes initialization followed by the Oseen iteration method for some iterations, then branching with the Newton method), or continuation methods (which advocate to increase steadily the Reynolds number) to improve and make the convergence process more robust. As we have already mentioned, the model examples considered in this chapter (see Section 2.5) arise in the regime of low Reynolds number, and we did not run into the need for such elaborated strategies.*

Remark 6. *In practice, we do not solve exactly (2.20), but the slightly modified version*

$$\left\{\begin{array}{rcll} -\nu\Delta\mathbf{u}^0 + \nabla p^0 &=& 0 & x \in \Omega, \\ \mathrm{div}(\mathbf{u}^0) + \varepsilon p^0 &=& 0 & x \in \Omega, \\ \mathbf{u}^0 &=& \mathbf{u}_{in} & x \in \Gamma_{in}, \\ \mathbf{u}^0 &=& 0 & x \in \Gamma, \\ \sigma(\mathbf{u}^0, p^0)\mathbf{n} &=& 0 & x \in \Gamma_{out}. \end{array}\right. \tag{2.23}$$

where ε *is a very small parameter (typically* $\varepsilon = 10^{-6}$*). The reason is that only the gradient of* p^0 *is involved in the system (2.20), which is not well-posed as a result:* p^0 *is only defined up to a constant; see Remark 3. In contrast, (2.23) is well-posed; the matrix associated to its resolution by the Finite Element method is positive definite, which allows to use efficient numerical linear algebra solvers; see e.g. [71, Chapter 4] about this approach. The same trick applies to (2.21). This is especially useful for* `FreeFem++` *resolutions, since direct sparse solvers are included.*

Choice of the Finite Element discretization.

When it comes to the numerical resolution of linear saddle point problems of the form (2.20) or (2.21), 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, (2.20) and

(2.21) 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 (2.20) and (2.21) are invertible. Details about numerical methods for the resolution of saddle point problems can be found in [58] or [62].

2.4.3 The augmented Lagrangian algorithm for equality-constrained problems

In order to drive the numerical resolution of (2.2), we rely on the augmented Lagrangian method, a basic sketch of which is provided; we refer to [111, §17.4] or [65] for detailed explanations.

The augmented Lagrangian algorithm transforms the constrained optimization problem (2.2) into the series of unconstrained problems (hereafter indexed by the superscript ^{n}):

$$\inf_{\Omega\in\mathcal{O}_{ad}} \mathscr{L}(\Omega, \ell^n, b^n), \tag{2.24}$$

where

$$\mathscr{L}(\Omega, \ell, b) = J(\Omega) - \ell G(\Omega) + \frac{b}{2} G(\Omega)^2 \tag{2.25}$$

In the definition of the augmented Lagrangian $\mathscr{L}$, 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 with this constraint in (2.2).

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

$$\ell^{n+1} = \ell^n - bG(\Omega^n) \tag{2.26}$$

We again refer to [111] 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 (2.2).

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

- **Initialization:** Start from an initial shape Ω^0 and coefficients ℓ^0 and b^0.
- **For** $n = 0, \ldots$ **until convergence**
  - choose a descent direction $\boldsymbol{\theta}^n$ for $\Omega \mapsto \mathscr{L}(\Omega, \ell^n, b^n)$,
  - take τ^n small enough so that $\mathscr{L}((\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n), \ell^n, b^n) < \mathscr{L}(\Omega^n, \ell^n, b^n)$, 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 $\mathscr{L}$ according to (2.26) and (2.27).

In the implementation, a variable coefficient b was chosen and updated using this rule

$$b^{n+1} = \begin{cases} \alpha b^n & \text{if } b < b_{\text{target}}, \\ b^n & \text{otherwise;} \end{cases} \tag{2.27}$$

In other terms, starting from a 'small' value b^0, the penalty b is increased by a user-defined factor $\alpha > 1$ during the first iterations of the optimization process, until the maximum, 'large' value b_{target} is reached: this smooth increase of b urges the optimized domain to fulfill the constraint in an increasingly stringent way in the course of the optimization process; see Section 2.5 for the actual values used in our implementation.

Notice also that the penalty coefficient b^n is multiplied by a user-defined constant $\alpha > 1$ during the first 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 to enforce

the constraint $G(\Omega) = 0$; this guarantees a better conditioning of (2.24) with respect to the naive quadratic penalty method (featuring only the first and last terms in the definition of $\mathscr{L}$ in (2.24) with $\ell^n = 0$). Choosing a variable b is not required for convergence but we noticed that it speeds up convergence.

2.4.4 Mesh-related issues

Assume for one moment that a descent direction $\boldsymbol{\theta}^n$ for (2.2) and a descent step τ^n have been found at the n^{th} iteration of the procedure described in Section 2.4.1; 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:

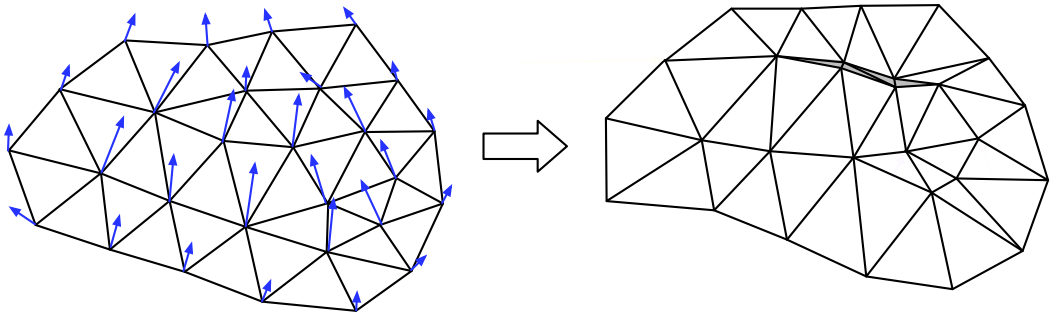
$$\mathbf{x}_i^n \mapsto \mathbf{x}_i^{n+1} := \mathbf{x}_i^n + \tau^n \boldsymbol{\theta}^n(\mathbf{x}_i^n), \quad i = 1, ..., I^n, \tag{2.28}$$

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 as $\mathcal{T}^n$, but their vertices are relocated according to $\boldsymbol{\theta}^n$.

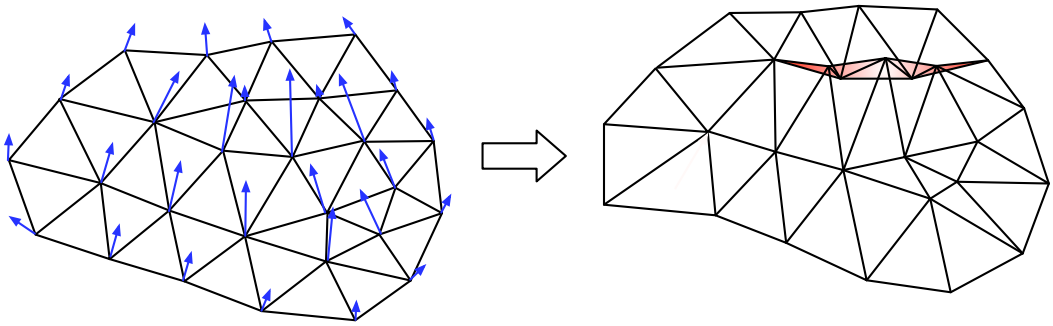
Unfortunately, this simple procedure is likely to give rise to very stretched (i.e. almost flat) elements within a few iterations. This is problematic since the accuracy of the resolution of PDE with the Finite Element method greatly depends on the *quality* of the elements in the mesh, i.e. on their being close to equilateral [48]. It may also happen that the mesh becomes overlapping in the course of the deformation; see Figure 2.9 for an illustration of such configurations.

Hence, the numerical resolution of (2.1) may become very inaccurate (not to say impossible) as the computational mesh is successively deformed, causing the whole 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 (2.28)), only the values of $\boldsymbol{\theta}^n$ on the boundary Γ^n



(a)



(b)

Figure 2.9 – *Examples of (a) a mesh getting very stretched (gray elements); (b) a mesh developing overlaps (red elements) in the course of its deformation.*

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 internal values of $\boldsymbol{\theta}^n$ may be chosen freely, in a way that makes $\mathcal{T}^{n+1}$ of good quality insofar as possible, as we describe in the next Section 2.4.5.

- When the quality of the mesh becomes poor, i.e. in our context when the volume of one of its elements becomes very small, i.e.

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

where $\varepsilon_{\text{mesh}}$ is a user-defined parameter (see [66] Chap. 18 for more details, in particular about other possible quality measures of a mesh, which could be easily implemented in FreeFem++.), a *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 moved,

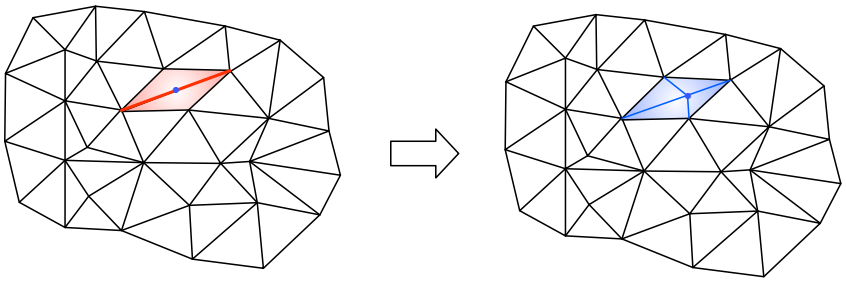
as long as the overall quality of the mesh is improved. See Figure 2.10 for an illustration of these operations.

From the practical implementation viewpoint, this complex series of operations is conveniently carried out owing to the `adaptmesh` command in `FreeFem++`.

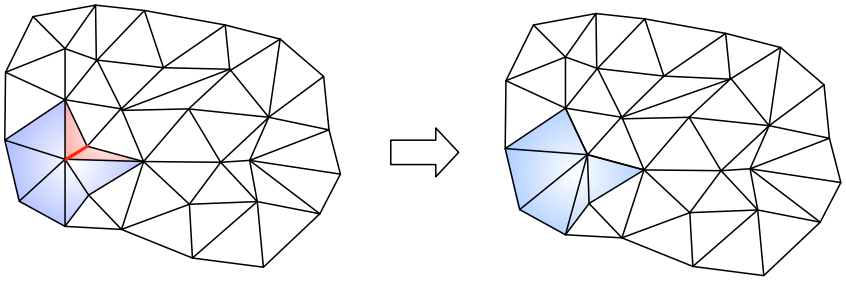
2.4.5 Extension-regularization of the shape gradient

As we have seen, our optimization procedure amounts to a series of minimizations of functionals of the form (2.24), which we generically denote by $F(\Omega)$ in the present section. We have seen in Section 2.3 that a natural candidate for a descent direction is

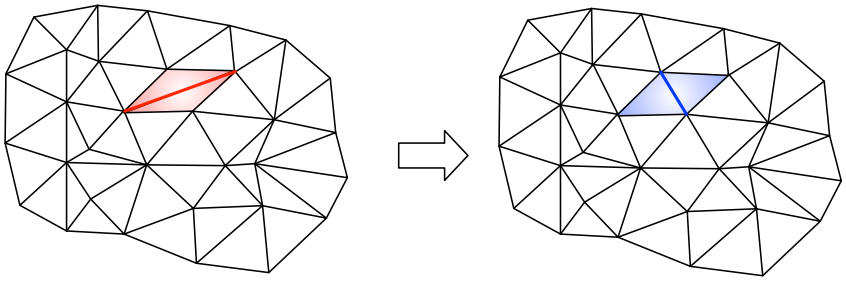
$$\boldsymbol{\theta} = -\phi\, \mathbf{n}, \tag{2.29}$$



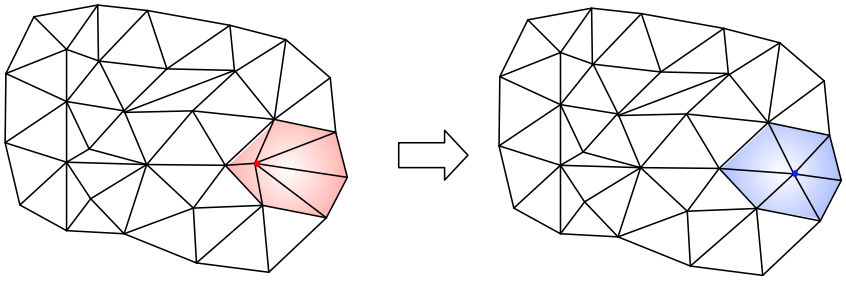
(a)



(b)



(c)



(d)

Figure 2.10 – *Illustrations of the remeshing operations described in Section 2.4.4: (a) splitting of a 'long' edge; (b) collapse of the two endpoints of a 'short' edge; (c) swap of the connectivities of a configuration of two ill-shapes triangles; (d) relocation of one vertex.*

where the scalar function $\phi : \Gamma \to \mathbb{R}$ is the $L^2(\Gamma)$-shape gradient of $F(\Omega)$, which is identified from the shape derivative of this functional via (2.16).

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

(i) Strictly speaking, (2.29) 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 (2.28).

(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 artifacts when it comes to the mesh procedure (2.28); see for instance [108] §6.2.4. It is therefore often desirable to smooth the velocity field $\boldsymbol{\theta}$ on Γ before performing (2.28).

The popular *extension-regularization* procedure provides alternative ways to calculate a descent direction $\boldsymbol{\theta}$ for $F(\Omega)$ from the knowledge of the shape derivative $F'(\Omega)(\boldsymbol{\theta})$ while overcoming both difficulties; see e.g. [38, 51, 57]. The basic idea consists in identifying a shape gradient for $F(\Omega)$ from its shape derivative $F'(\Omega)(\boldsymbol{\theta})$ (see (2.16)) 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 for each test function $\boldsymbol{\psi} \in V$,

$$(\boldsymbol{\theta}, \boldsymbol{\psi})_V = J'(\Omega)(\boldsymbol{\psi}) = \int_\Gamma \phi\, \boldsymbol{\psi} \cdot \mathbf{n}\, ds. \tag{2.30}$$

Doing this ensures that:

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

which together with (2.17) 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

$$\begin{aligned} \forall \boldsymbol{\theta}, \boldsymbol{\psi} \in V, \ \ (\boldsymbol{\theta}, \boldsymbol{\psi})_V = \gamma \int_\Omega Ae(\boldsymbol{\theta}) : e(\boldsymbol{\psi}) \, dx \\ + (1-\gamma) \int_\Gamma \nabla_\Gamma \boldsymbol{\theta} \cdot \nabla_\Gamma \boldsymbol{\psi} \, ds. \end{aligned} \tag{2.31}$$

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

- The first term in (2.31) is inspired by the linearized elasticity equations. Here, A is the Hooke's law, acting on symmetric matrices e with size $d \times d$,

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

  where λ and μ are the Lamé coefficients of the fictitious elastic material and A is the operator associated to Hooke's law :

  $$\begin{array}{lcl} A : S_n(\mathbb{R}) & \to & S_n(\mathbb{R}) \\ M & \mapsto & AM = 2\mu M + \lambda \, \mathrm{tr}(M) I \end{array}$$

  This choice - which is common in meshing [18, 56] to help in keeping a mesh with fine quality - is motivated by the intuition that elastic displacements tend to induce little shear, thus avoiding mesh self-intersections.

- The second term in (2.31) corresponds to the Laplace-Beltrami operator on Γ; its role is to enforce the smoothness of the descent direction $\boldsymbol{\theta}$ on Γ.

With these definitions at hand, the desired 'regularized' shape gradient $\boldsymbol{\theta}$ is calculated by solving (2.30) with a standard Finite Element method on a mesh of Ω.

Remark 7. *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 8. *A perhaps more natural idea consists in choosing*

$$V = \left\{ \mathbf{v} \in H^1(\Omega)^d, \;\; \mathbf{v}|_{\Gamma_{in} \cup \Gamma_{out}} = 0 \right\},$$

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, (2.30) amounts to solving the regularizing, elliptic system:

$$\left\{ \begin{array}{rcll} -\gamma \Delta \boldsymbol{\theta} + \boldsymbol{\theta} & = & 0 & x \in \Omega, \\ \boldsymbol{\theta} & = & 0 & x \in \Gamma_{in} \cup \Gamma_{out}, \\ \gamma \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{n}} & = & -\phi \mathbf{n} & x \in \Gamma. \end{array} \right. \tag{2.32}$$

However easy to implement, this choice is less efficient than (2.31) insofar as it does not show the same efficiency in preventing the emergence of stretched elements; see the example in Section 2.5.3 about this point.

2.4.6 Calculation of the curvature

Most of the numerical methods involved in the resolution of the shape optimization problem (2.2) imply the calculations of the normal vector $\mathbf{n}$ and the 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 [66], 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 2.11.

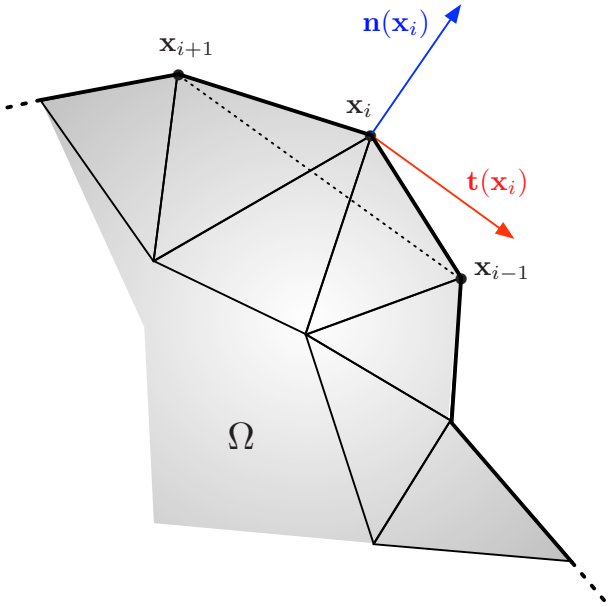


Figure 2.11 – *Estimation of the tangent and normal vectors to $\partial\Omega$ from the data of a triangular mesh.*

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 $\mathbf{t}(\mathbf{x}_i)$ in 2D

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

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

$$r(\mathbf{x}_i) = \frac{1}{4}\left(\frac{\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}} \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}}{-\mathbf{n}(\mathbf{x}_i) \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}} + \frac{\overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}} \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}}{-\mathbf{n}(\mathbf{x}_i) \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}}\right), \tag{2.33}$$

and the curvature $\kappa(\mathrm{x}_i)$ at $\mathbf{x}_i$ is simply $\kappa(\mathrm{x}_i) = \frac{1}{r(\mathbf{x}_i)}$ if none of the denominators featured in (2.33) equals 0 (it is set to 0 otherwise).

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 $(\mathbf{u}^n, p^n)$ to the Navier-Stokes system (2.1) on the mesh $\mathcal{T}^n$ of Ω^n by using the material in Section 2.4.2.
   (ii) Calculate the solution $(\mathbf{v}^n, q^n)$ to the adjoint system (2.11) or (2.13) on Ω^n.
   (iii) Calculate the $L^2(\Gamma^n)$ shape gradient ϕ^n of $\Omega \mapsto \mathscr{L}(\Omega, \ell^n, b^n)$ by using Theorem 2.
   (iv) Infer a descent direction $\boldsymbol{\theta}^n$ for $\Omega \mapsto \mathscr{L}(\Omega, \ell^n, b^n)$ by solving (2.30) (2.31) on the mesh $\mathcal{T}^n$.
   (v) Find a descent step τ^n such that
   $$\mathscr{L}((\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n), \ell^n, b^n) < \mathscr{L}(\Omega^n, \ell^n, b^n) \tag{2.34}$$
   (possibly up to a small tolerance)
   (vi) Move the vertices of $\mathcal{T}^n$ according to τ^n and $\boldsymbol{\theta}^n$:
   $$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \tau^n \boldsymbol{\theta}^n(\mathbf{x}_i^n) \tag{2.35}$$
   - If the resulting mesh is invalid, go back to step (v), and use a smaller value for τ^n,
   - Else, the positions (2.35) 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 2.4.4.
   (viii) Update the augmented Lagrangian parameters according to (2.26).
3. **Ending criterion.** Stop if
   $$\|\boldsymbol{\theta}^n\|_{L^2(\Gamma^n)} < \varepsilon_{\text{stop}}$$
   Return Ω^n.

Figure 2.12 – *Synthetic description of the optimization loop*

2.4.7 Algorithmic description of the implemented method

We are now ready to provide a precise sketch of the shape optimization algorithm arising from the previous considerations, see Figure 2.12. The brief account below follows exactly the steps of the file `main.edp` of the (commented) supplied code.

2.5 Numerical illustrations

In this section, we present five two-dimensional applications of the numerical algorithm presented in Section 2.4.7. The geometric configurations associated to these examples are represented in Figure 2.13, and the parameters used in the different test cases (initial parameters of the augmented Lagrangian algorithm, target volume, etc.) are reported in Table 2.1. The approximate CPU time when running each example on a workstation with an Intel Core i5-7600T @ 2.80GHz CPU in `FreeFem++` is indicated in Table 2.1.

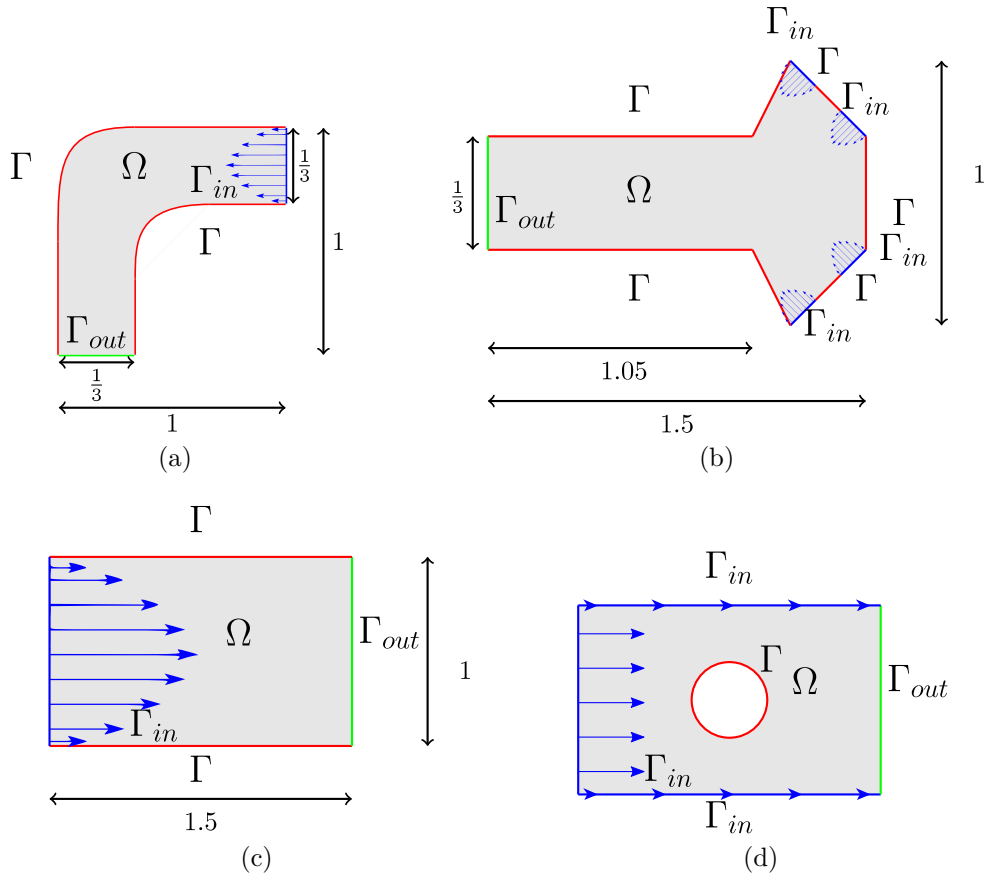


Figure 2.13 – *Settings of the five test cases discussed in Section 2.5; (a) the bend, discussed in Section 2.5.1, (b) the ramified structure of Sections 2.5.2 and 2.5.3, (c) the straight pipe with one inlet, one outlet where a least-square criterion is considered, as studied in Section 2.5.5, (d) the dissipated energy minimization example of Section 2.5.6.*

	$\frac{V_T}{V_0}$	$\frac{P_T}{P_0}$	γ	τ^0	$\varepsilon_{\text{stop}}$	ℓ^0	b_0	b_{target}	Re	Execution duration
Case 1	1		10^{-2}	10^{-2}	10^{-2}	0	1	10^1	200	3mn24s
Case 2	1		10^{-2}	10^{-2}	5×10^{-3}	0	10^{-1}	10^1	200	10mn8s
Case 3			10^{-2}	10^{-2}		0	0	0	200	4mn28s
Case 4	1		1	3×10^{-3}	2×10^{-2}	15	10^1	10^2	200	12mn7
Case 5		0.97	10^{-2}	10^{-2}	10^{-2}	0	10^2	10^2	200	6mn43s

Table 2.1 – *Parameters used for the numerical examples of Section 2.5. From left to right: desired volume over initial volume, desired perimeter over initial perimeter, regularization parameter, initial gradient step, stopping criterion, initial Lagrange multiplier, initial value of b, target value of b, Reynolds number and execution time on a recent workstation (as of 2018).*

2.5.1 Minimization of the dissipated energy in a bending pipe

Our first benchmark example is concerned with the optimization of the shape of a pipe with orthogonal inlet and outlet, as depicted in Figure 2.13 (a); see for instance [49, 30]. In a nutshell, this test case answers the question:

> "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

$$\mathbf{u}_{\text{in}}(x_1, x_2) = \left((1 - x_2) \left(\frac{2}{3} - x_2 \right), 0 \right) \tag{2.36}$$

Starting from the initial shape Ω^0 represented in Figure 2.15 (top), we minimize the work of viscous forces, i.e. $J(\Omega) = E(\Omega)$, as defined by (2.3), under the volume constraint $\text{Vol}(\Omega) = V_T$, where $V_T = \text{Vol}(\Omega^0)$, i.e. the target volume is that of Ω^0.

The results are displayed on Figure 2.15, and the associated convergence histories are included in Figure 2.14. The dissipated viscous energy decreases by roughly 25% during the process, and as expected, the optimized design looks like a straight pipe. It is worth mentioning that theoretical arguments in [83] support this observation for a very close model.

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.

2.5.2 Minimization of the dissipated energy in a ramified structure with volume constraint

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 [53].

The situation is that of Figure 2.13 (b), where incoming parabolic profiles are imposed on each component of Γ_{in}. More precisely, Γ_{in} is

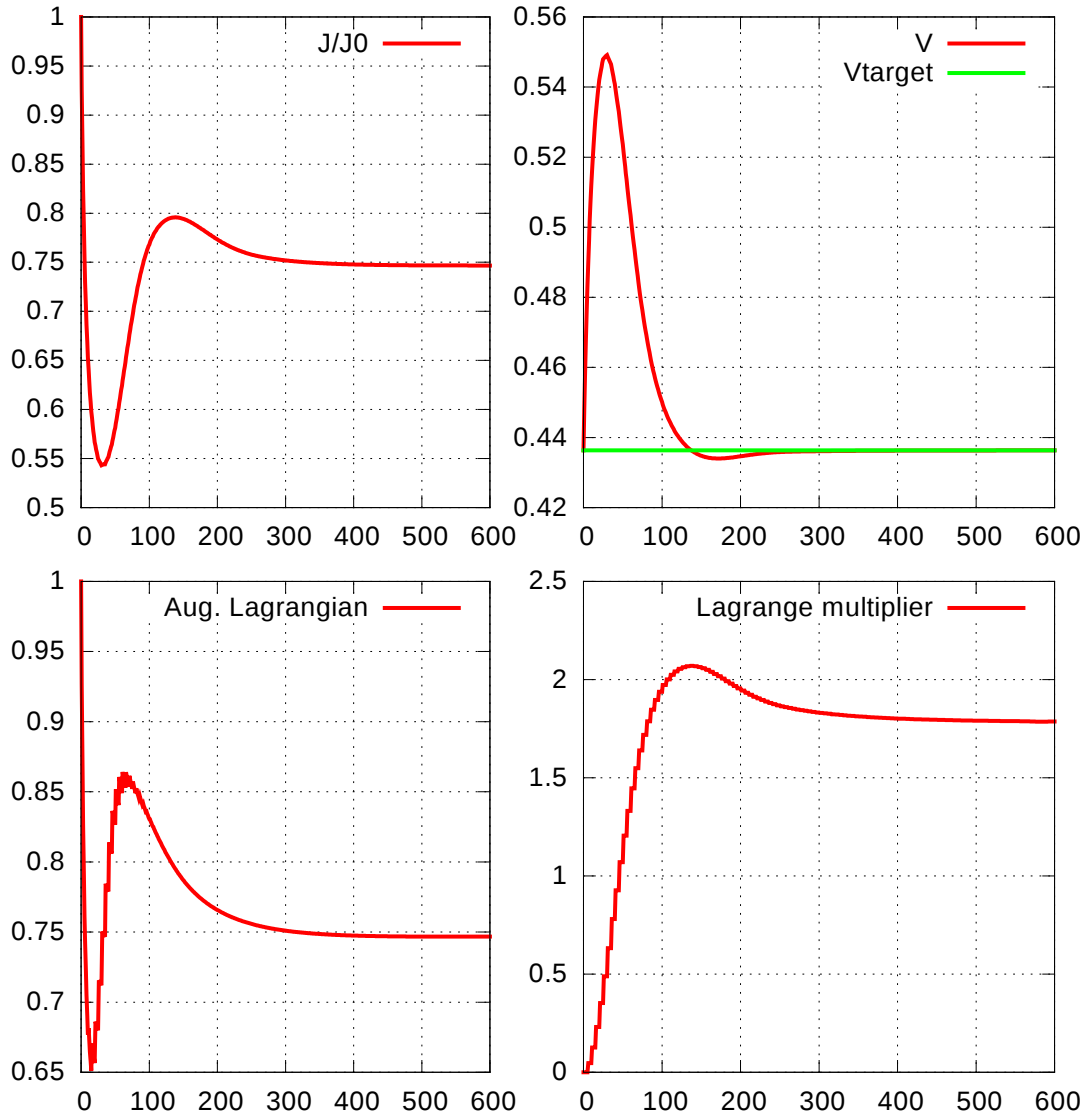


Figure 2.14 – *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in the bend optimization example of Section 2.5.1*

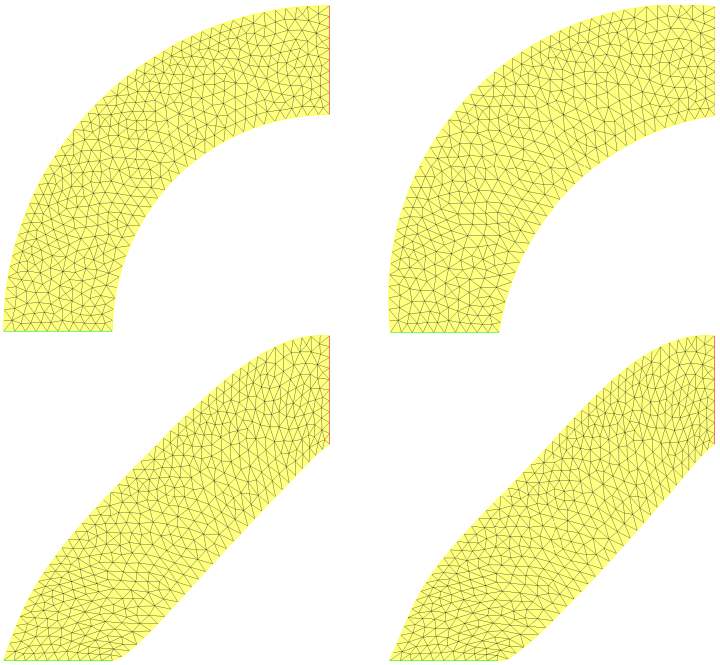


Figure 2.15 – *Intermediate shapes* Ω^n *obtained in the bend optimization example of Section 2.5.1 at iterations (from top to bottom)* $n = 0, 5, 100$ *and* 500.

the reunion of four disjoint line segments; for any of these segments, let us denote by (x_1^A, x_2^A) and (x_1^B, x_2^B) the two ending points, which are assumed to be distributed counterclockwise on Γ_{in}. The imposed inlet flow on the considered segment is then defined by:

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

where $s = \frac{x_1^A - x_1}{x_1^A - x_1^B} = \frac{x_2^A - x_2}{x_2^A - x_2^B}$ so that in particular $\mathbf{u}_{\text{in}}$ is oriented toward inside Ω: $\mathbf{u}_{\text{in}} \cdot \mathbf{n} \leq 0$ on Γ_{in}.

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

The results are presented in Figure 2.16, and the associated convergence histories are those in Figure 2.17. Interestingly enough, ramifications appear in the course of the iterations and the optimized shape is much smoother than the initial one. These results are also in accordance with those obtained in [53].

This example shows large mesh deformations, which justifies the

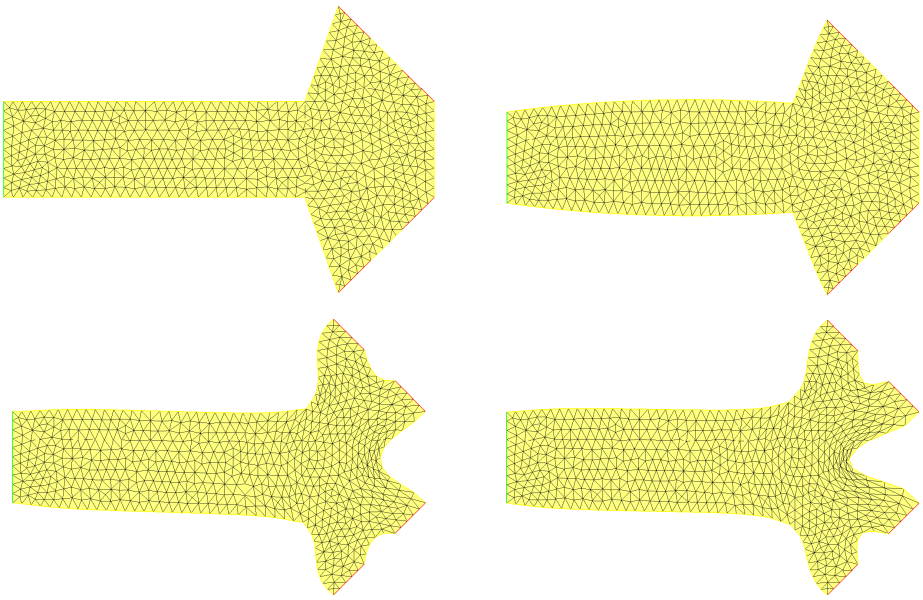


Figure 2.16 – *From top to bottom, successive shapes* Ω^n *at iterations* $n = 0, 5, 240, 1000$ *in the dissipated energy minimization example in a ramified structure with volume constraint of Section 2.5.2.*

importance of using a good extension-regularization procedure, such as that introduced in section 2.3.

2.5.3 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 2.5.2 (again, see Figure 2.17. The only difference with the latter is that we now impose a constraint on the perimeter of shapes: $\mathrm{Per}(\Omega) = P_T$, with $P_T = 0.97\,\mathrm{Per}(\Omega^0)$. The convergence histories of the computation are reported on Figure 2.18, and the shape at several intermediate stages is represented on Figure 2.19.

Let us emphasize the role of the regularizing parameter γ featured in the definition of the extension-regularization inner product (2.31). In this example (as in the previous ones), the $L^2(\Gamma)$ shape gradient of $E(\Omega)$ is not smooth in the vicinity of the transitions between parts of the boundary bearing different types of boundary conditions (that is, the transitions between Γ_{in}, Γ_{out} or Γ). Therefore, if no

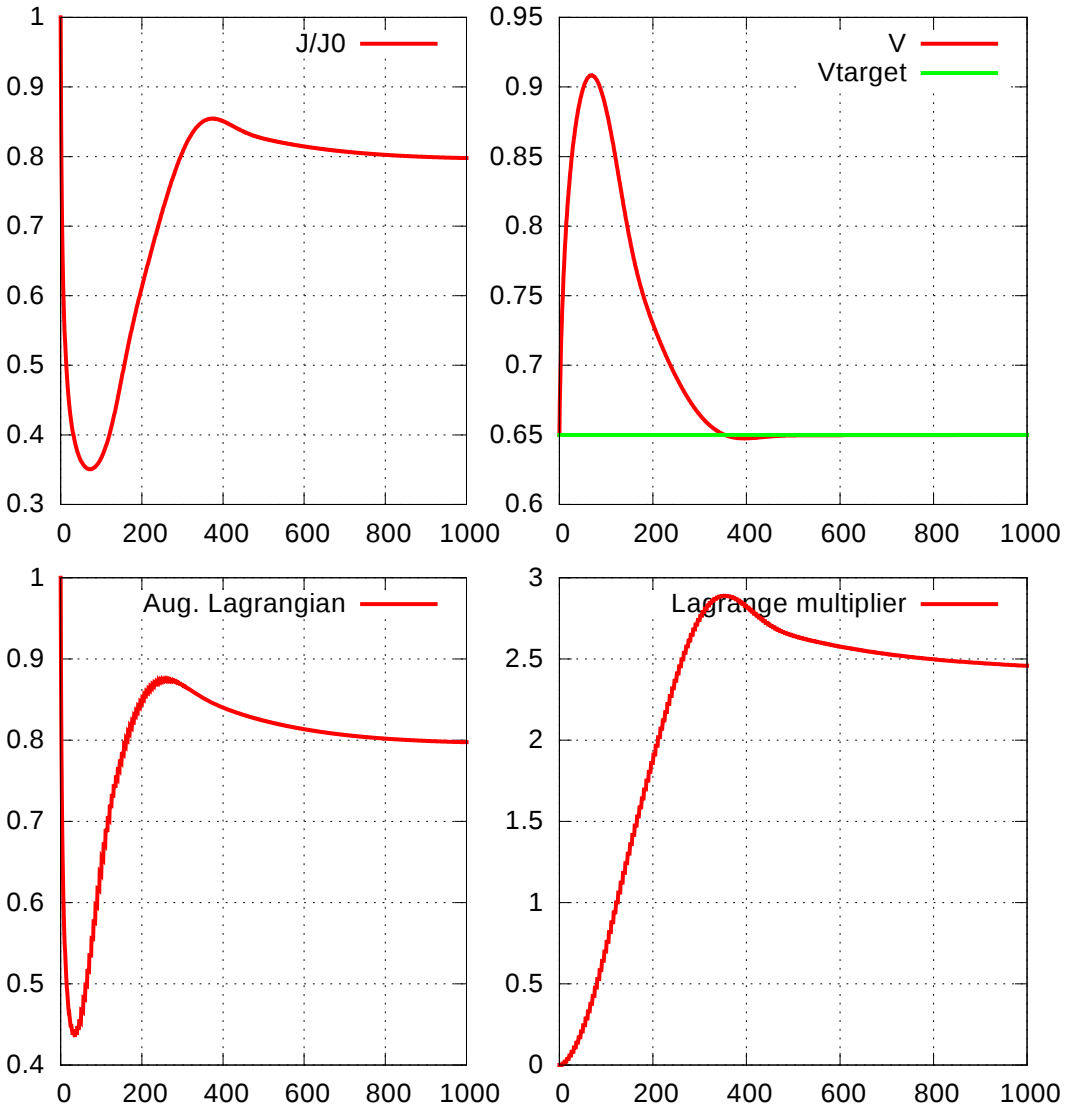


Figure 2.17 – *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in the dissipated energy minimization example in a ramified structure of Section 2.5.2.*

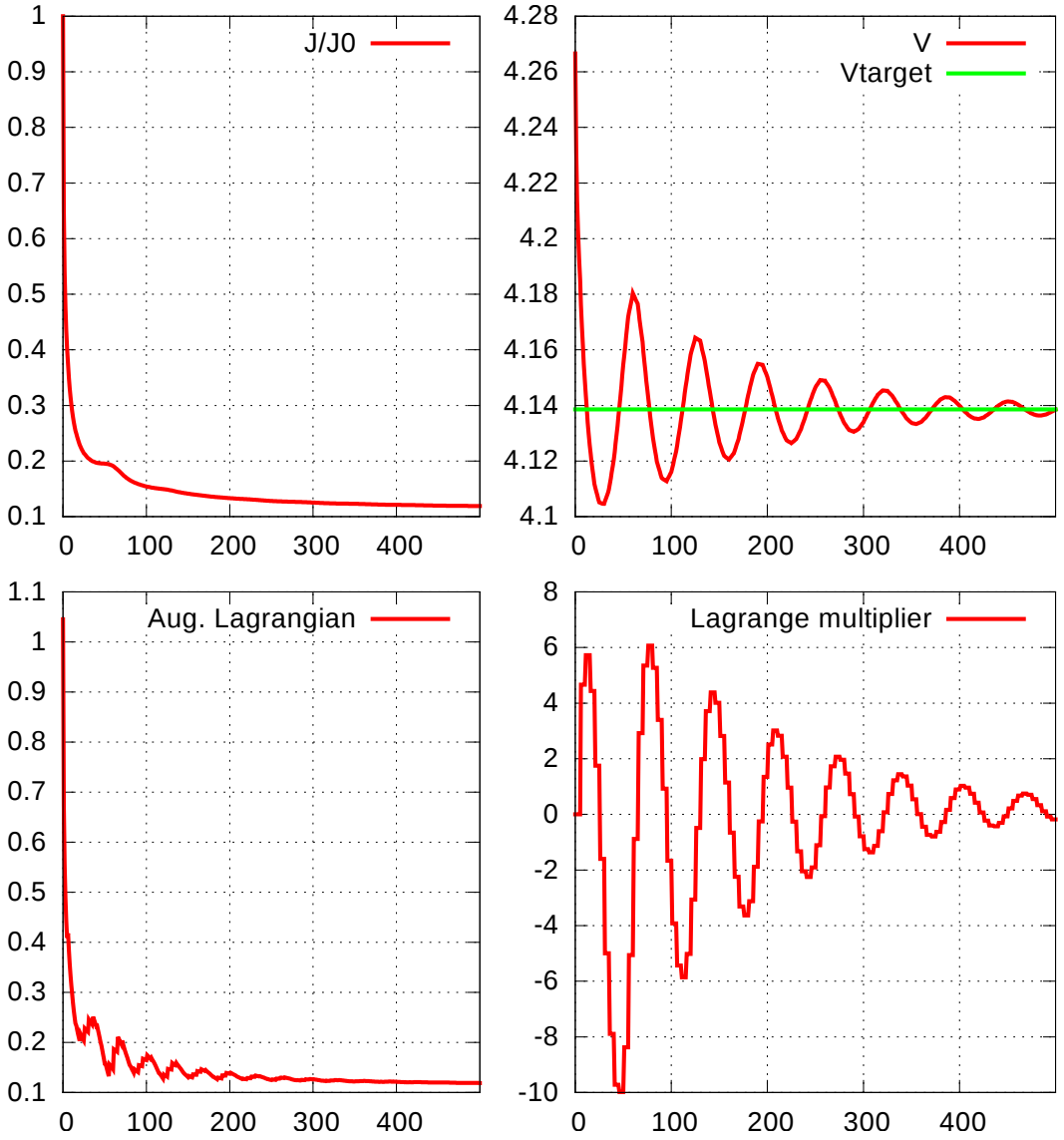


Figure 2.18 – *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Per}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in dissipated energy minimization in a ramified structure example of Section 2.5.3*

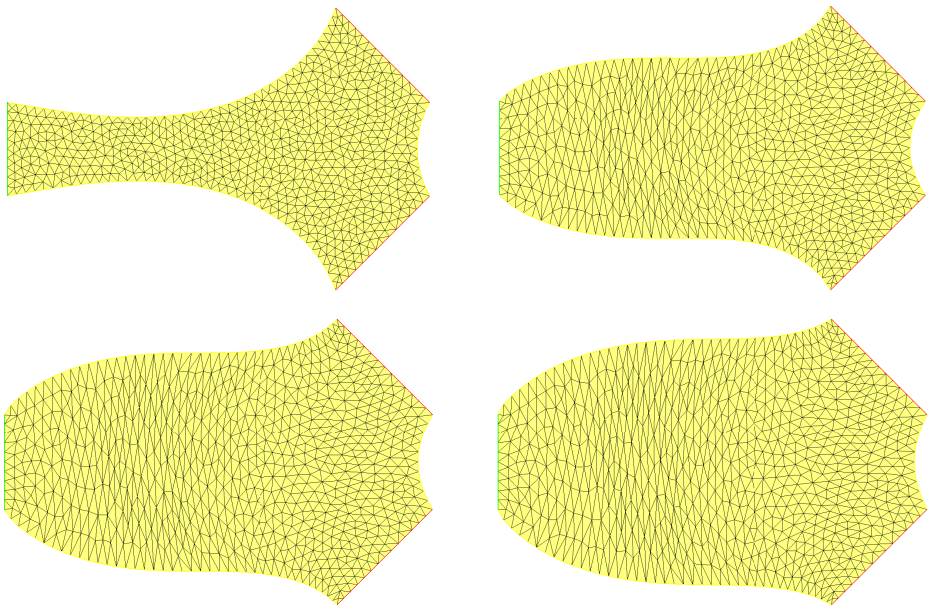


Figure 2.19 – *From top to bottom, successive shapes* Ω^n *at iterations* $n = 0, 100, 250, 500$ *in the energy dissipation example in a ramified structure with perimeter constraint of Section 2.5.3.*

regularization of this gradient is applied, mesh intersections appear within a few iterations in this region.This phenomenon is illustrated in Figure 2.20: all parameters retain the same values except for γ, which is changed to 1. The mesh irregularities are caused by an irregular shape gradient, and are not observed for $0 < \gamma < 1$.

2.5.4 Squared vorticity in a ramified structure

As with the test-case presented in as depicted in 2.5.3, the geometry of this test-case is presented in Figure 2.13 (b). The goal of this test-case is to answer the question

> "How to build a pipe with fixed perimeter that makes the flow as irrotational as possible ?"

The squared vorticity functional is related to the dissipated energy dissipation, see Remark 1. The inlet flow is given by the parabolic profile (2.36).

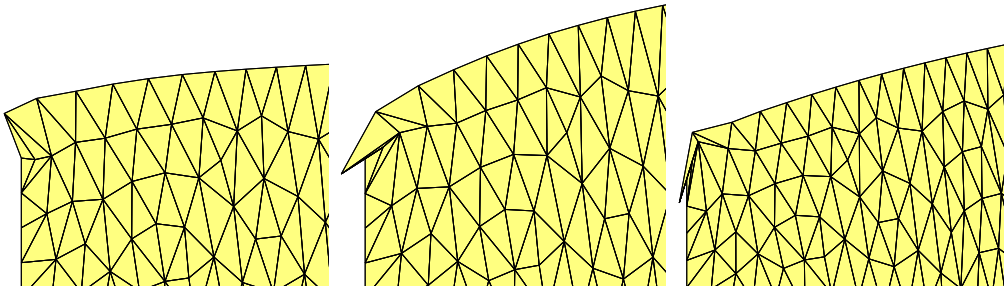


Figure 2.20 – *Left upper corner of intermediate shapes Ω^n obtained in the ramified structure example of Section 2.5.3 at iterations (from top to bottom) $n = 15, 60$ and 130.*

Starting from the initial shape Ω^0 represented in Figure 2.21 (top), we minimize the total squared vorticity, i.e. $J(\Omega) = C(\Omega)$, as defined by (2.5), under the perimeter constraint $\mathrm{Per}(\Omega) = 0.97 P_T$, where $P_T = \mathrm{Vol}(\Omega^0)$.

2.5.5 Minimization of the discrepancy with a reference velocity profile

Our third example considers pipes Ω in the situation depicted on Figure 2.13 (c), where the parabolic profile

$$\mathbf{u}_{\mathrm{in}}(x_1, x_2) = (x_2(1 - x_2), 0) \tag{2.37}$$

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:

$$D(\Omega) = \int_{\Gamma_{\mathrm{out}}} |\mathbf{u} - \mathbf{u}_{\mathrm{ref}}|^2 \, ds, \tag{2.38}$$

where the reference profile $\mathbf{u}_{\mathrm{ref}}$ is the velocity of the fluid, solution to (2.1), associated to the same incoming flow (2.37), on the reference domain Ω_{ref} defined by:

$$\Omega_{\mathrm{ref}} := \left\{ (x_1, x_2) \in \mathbb{R}^2,\ 0 \le x_1 \le \frac{3}{2},\ \ f_{\mathrm{ref}}(x_1) \le x_2 \le f_{\mathrm{ref}}(x_1) + 1 \right\}, \tag{2.39}$$

with

$$f_{\mathrm{ref}}(x_1) = \frac{1}{5} x_1 \left(\frac{3}{2} - x_1 \right).$$

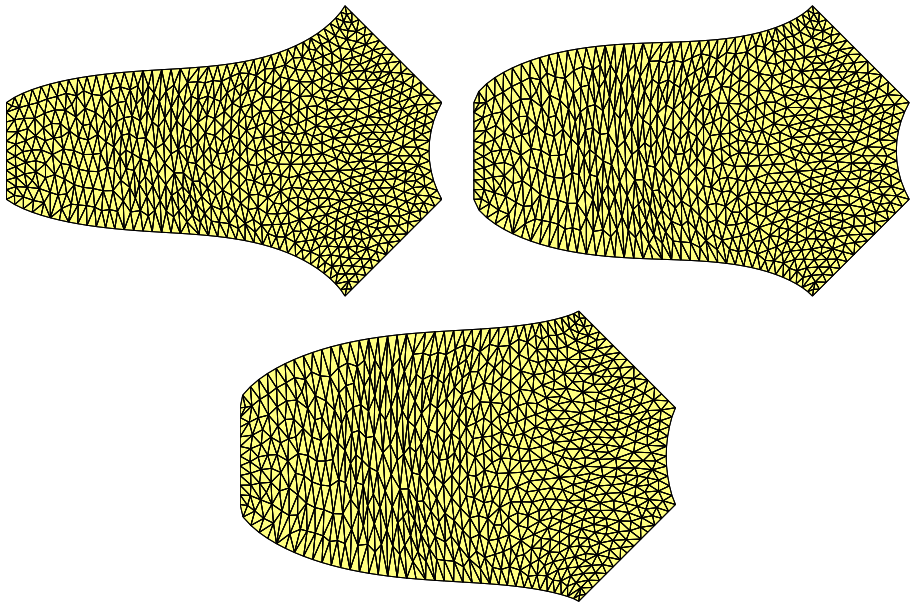


Figure 2.21 – *Intermediate shapes* Ω^n *obtained in the ramified structure example of Section 2.5.4 at iterations (from top to bottom)* $n = 200, 300$ *and* 474.

In other terms, in this test-case, the resolution of the optimization problem (2.2) aims to recover the shape of the reference domain Ω_{ref} from the sole knowledge of the velocity profile $\mathbf{u}_{\text{ref}}$ on Γ_{out}. By construction, (2.2) has at least one solution $\Omega = \Omega_{\text{ref}}$, since $D(\Omega_{\text{ref}}) = 0 \leq D(\Omega)$ for all $\Omega \in \mathcal{O}_{ad}$; however, this solution may not be unique. Note also that, obviously, other choices are possible as far as the reference profile $\mathbf{u}_{\text{ref}}$ is concerned, but depending on whether this choice is 'physical' or not, it may not be possible to realize the latter. The above choice of $\mathbf{u}_{\text{ref}}$ ensures that $D(\Omega)$ should be "small" upon convergence of the numerical resolution of (2.2).

The considered shape optimization problem in this case is unconstrained - i.e. no occurrence of the constraint $G(\Omega)$ appears in (2.2); from the numerical viewpoint, the framework of Section 2.4.7 reduces to a simple gradient algorithm: $b = \ell_0 = 0$.

Several intermediate shapes in the resolution of (2.2) and the reference shape Ω_{ref} are presented on Figure 2.26, and the corresponding one-dimensional profiles of the components u_1 and u_2 of the velocity $\mathbf{u}$ of the fluid on Γ_{out} are reported on Figure 2.25. These profiles get closer and closer to their reference counterparts as the

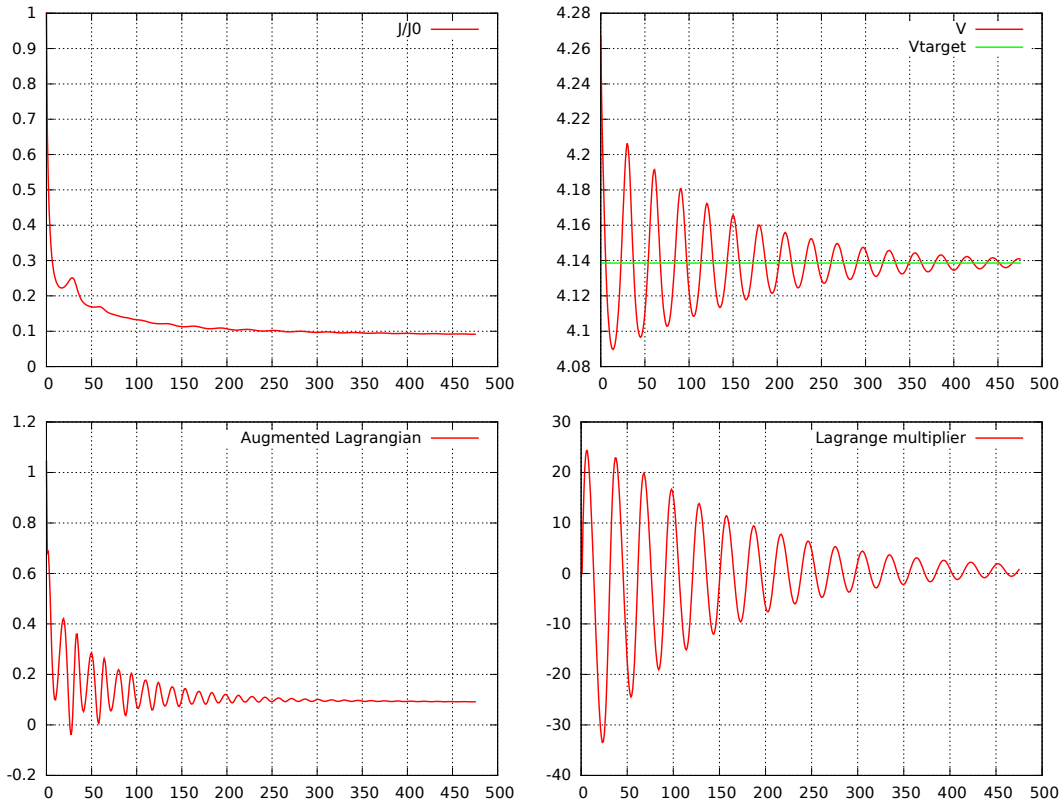


Figure 2.22 – *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Per}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in squared vorticity minimization in a ramified structure example of Section 2.5.4*

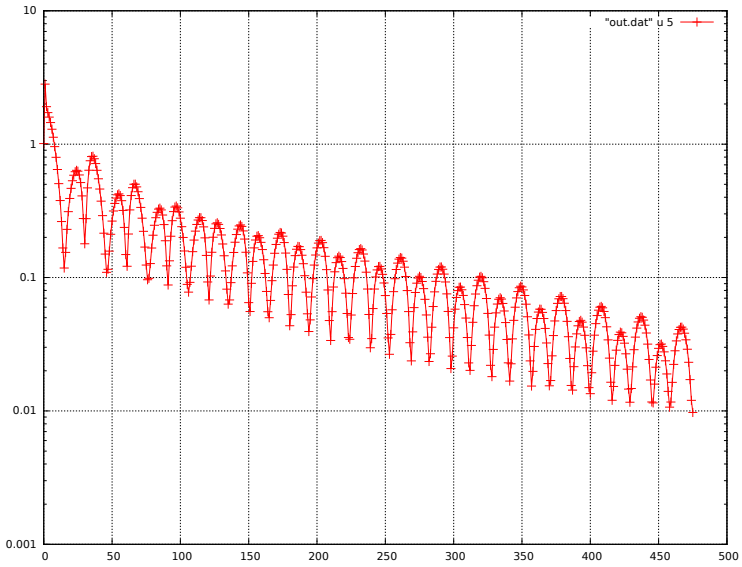


Figure 2.23 – *Stopping criterion in the minimization in a ramified structure example of Section 2.5.4*

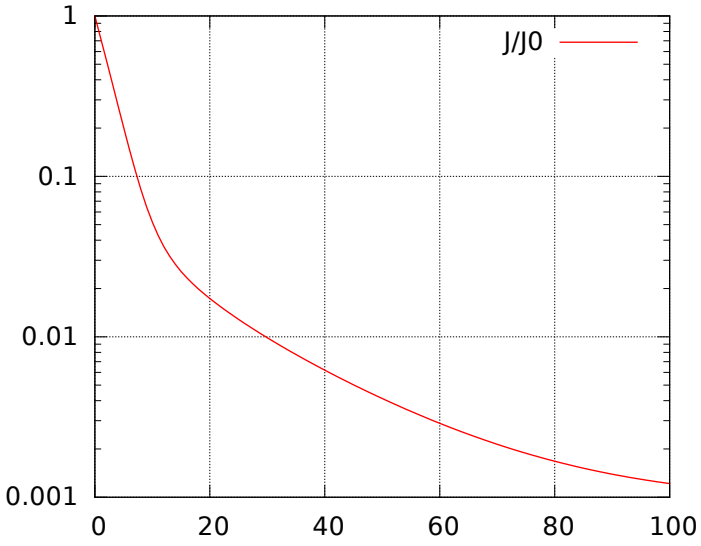


Figure 2.24 – *Convergence history of $J(\Omega)$ in the least-square criterion minimization example of Section 2.5.5.*

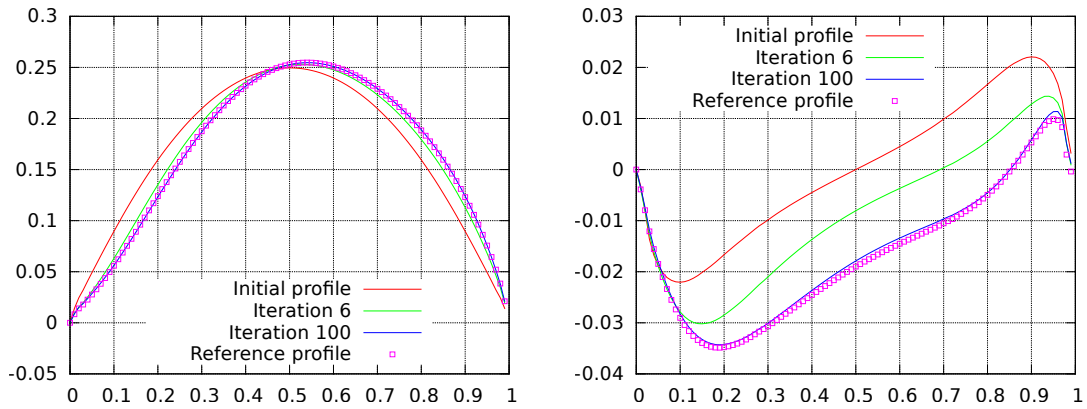


Figure 2.25 – *One-dimensional profiles of (top) u_1 and (bottom) u_2 on Γ_{out} at several stages in the example of Section 2.5.5.*

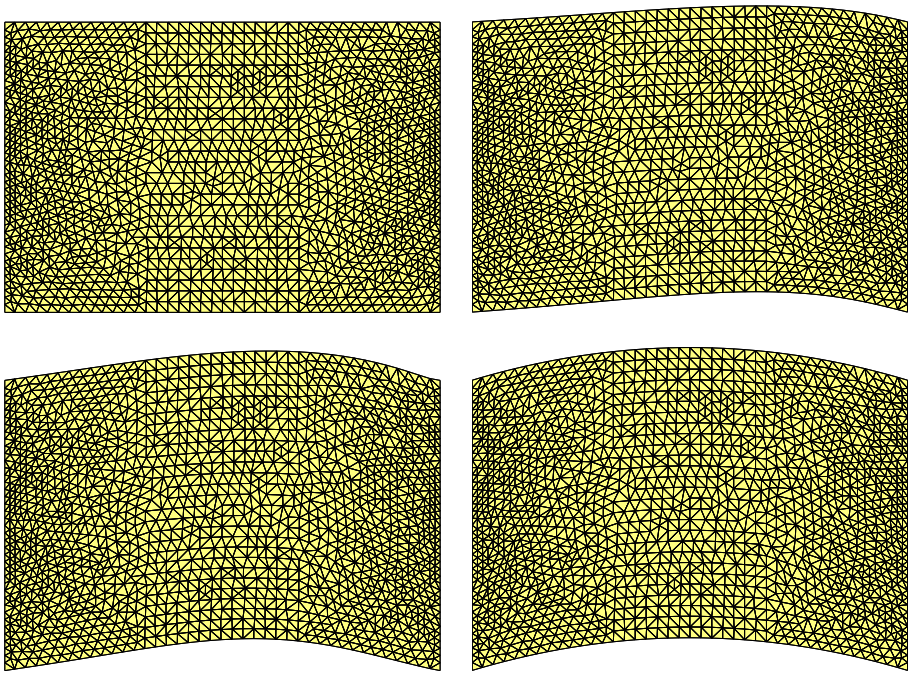


Figure 2.26 – *From top left to bottom right, successive shapes Ω^n at iterations $n = 0, 6, 100$ and reference shape in the least-square criterion minimization example of Section 2.5.5.*

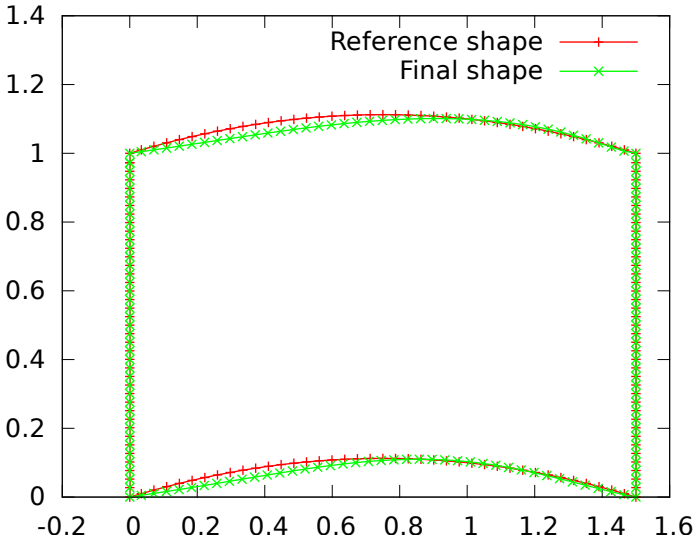


Figure 2.27 – *Boundary of the reference shape and shape at convergence in the least-square criterion minimization example of Section 2.5.5.*

algorithm reaches convergence, and the value of the minimized least-square criterion $D(\Omega)$ decreases to roughly 0.1% of its initial value, see the convergence history in Figure 2.24. A close inspection of Figure 2.27 reveals that the optimized shape Ω does not match with the reference Ω_{ref}. This example as well as other, unreported tests carried out using different reference shapes Ω_{ref} show that the latter is matched properly by the result Ω from the optimization process in the region close to the boundary Γ_{out}, but that both shapes differ significantly far from this region. This can be explained by the fact that at low Reynolds number, the upstream flow has little influence on the downstream flow; see [40] for a related observation.

Assuming that there exist more than one shape Ω satisfying $D(\Omega) = 0$, an interesting problem would be to find one satisfying this property while minimizing the dissipated energy $E(\Omega)$ given by (2.3), namely:

$$\min_{\Omega \in \mathcal{O}_{\text{ad}}} E(\Omega) \text{ s.t. } D(\Omega) = 0.$$

2.5.6 Energy dissipation around an obstacle

In our last example, depicted on Figure 2.13 (d), 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. $J(\Omega) = E(\Omega)$ with the volume constraint $\mathrm{Vol}(\Omega) = V_T$, $V_T = \mathrm{Vol}(\Omega^0)$.

A very similar version of this problem is considered in [121] and [31] in the context of Stokes flows. The same problem was later investigated using more modern topology optimization techniques in [30].

In the model situation discussed here, we impose a horizontal flow on Γ_{in}, namely

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

and no-slip boundary conditions are prescribed on the boundary of the obstacle. The convergence histories are presented on Figure 2.28. The resulting shape (see Figure 2.29, bottom) is roughly similar to those obtained in references [31, 30], having the visual aspect of a sharp rugby ball.

Finally, 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 in order to avoid the collapse of the obstacle, which would result in an invalid mesh. From a practical point of view, this choice relies on a few trials on very coarse meshes, assuming those meshes are not too coarse so that they not mask the underlying tendencies of the algorithm.

2.5.7 Effect of the mesh refinement on numerical results

In this section, we investigate numerically the sensitivity of the results obtained for Test-case 1 in Section 2.5.1 with respect to the mesh size. We define the mesh-size as the edge-size of a uniform mesh. With this definition, we take $h = \frac{1}{30}$, $h = \frac{1}{50}$ and $h = \frac{1}{70}$ and compare

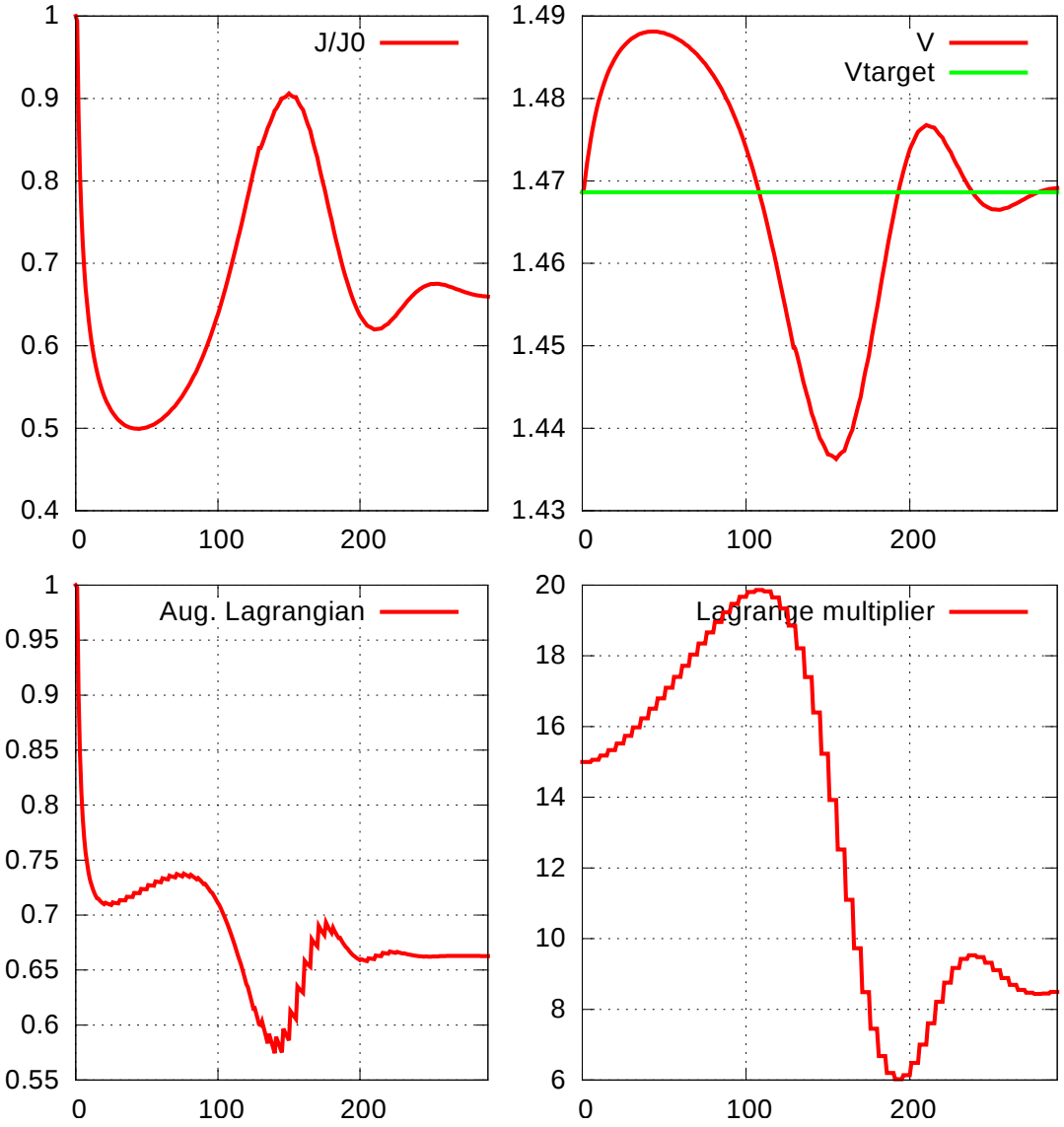


Figure 2.28 – *Convergence histories of (from left to right, top to bottom)* $J(\Omega)$, $\mathrm{Vol}(\Omega)$, $\mathscr{L}(\Omega, \ell, b)$ *and* ℓ^n *in the dissipated energy minimization example of Section 2.5.6*

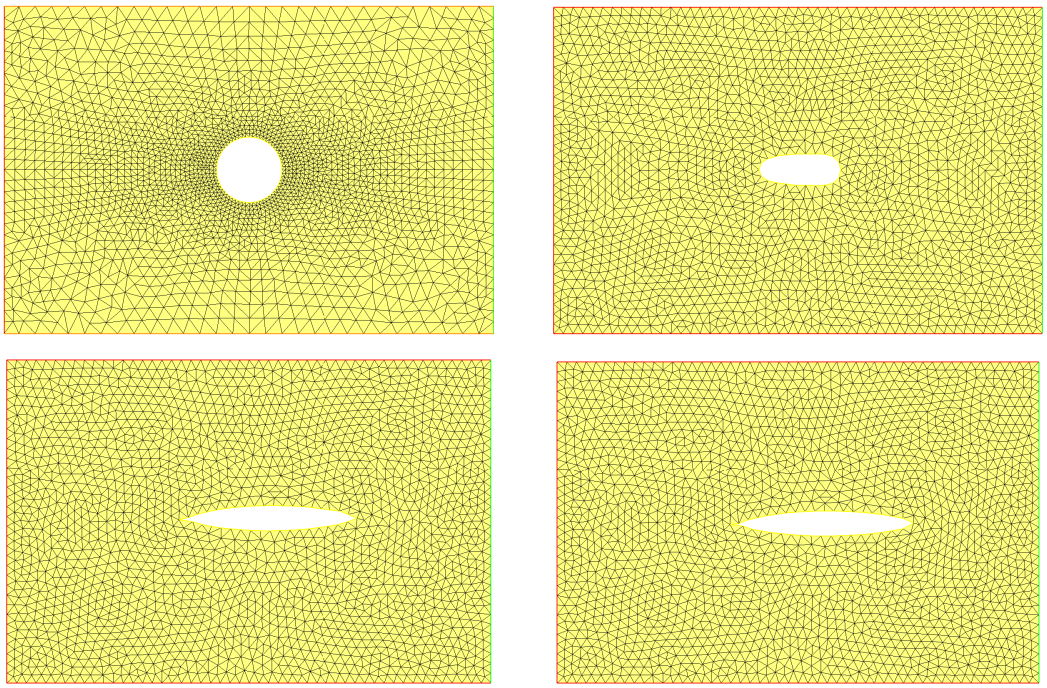


Figure 2.29 – *From top to bottom, successive shapes* Ω^n *at iterations* $n = 5,\ 100,\ 650$ *in the dissipated energy minimization example of Section 2.5.6.*

the results obtained in 2.5.1. In order to keep comparable results, since Test-case 1 can be run without it, we disabled adaptive mesh refinement during the optimization procedure. It is not an absolute necessity, we could instead have enabled adaptive refinement with different prescribed mesh-sizes. All parameters are taken as in Table 2.1.

Figures 2.30 and 2.32 indicate that at convergence, both the functional and the shape converge as $h \to 0$. Figure 2.31 shows that the cost functional is quite similar during the whole optimization process. At convergence, the only notable difference is in the detail area of Figure 2.30, which can be associated to an important local deformation, suggesting that the error is geometrical. Using adaptive refinement may allow to avoid these differences.

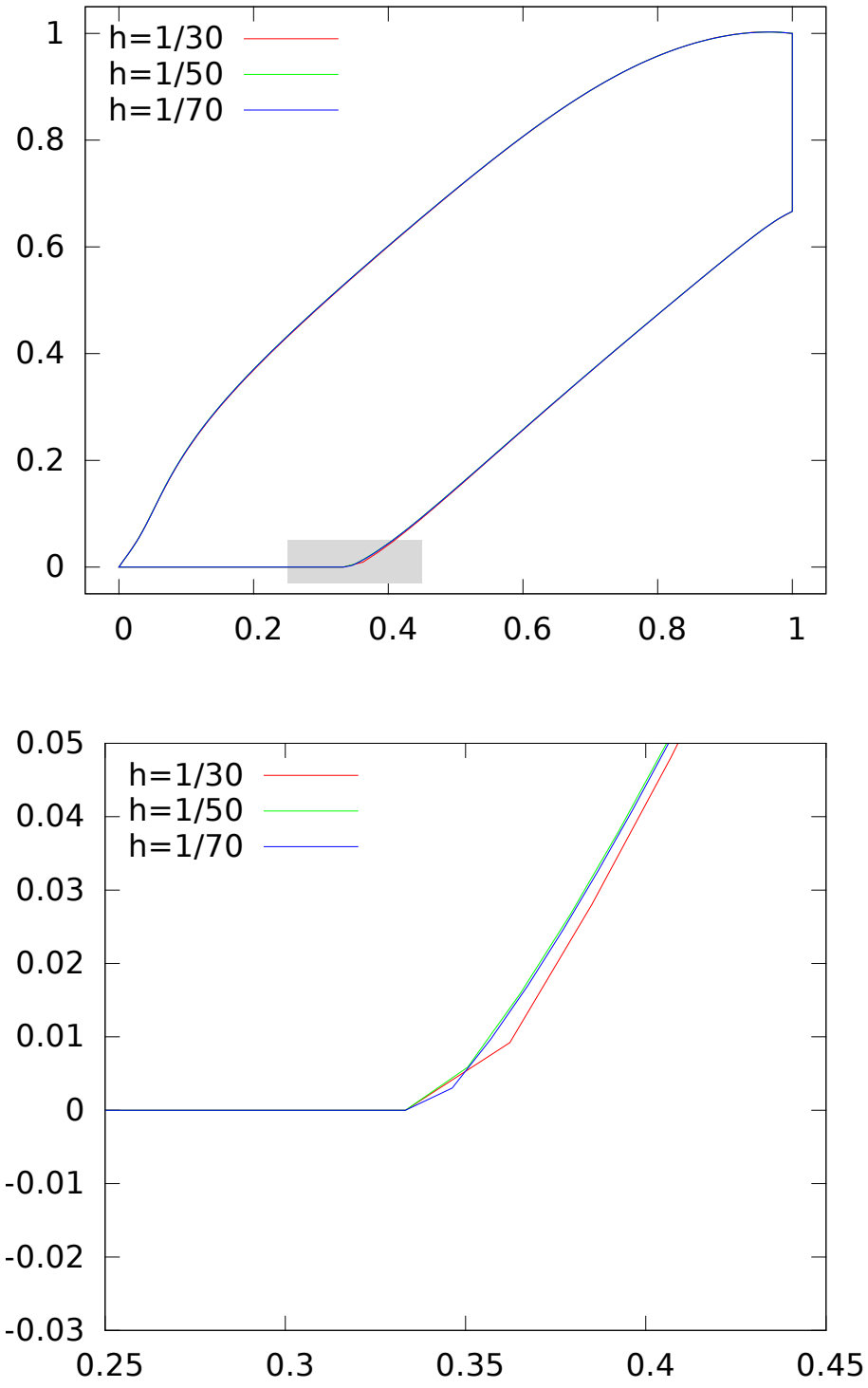


Figure 2.30 – *Boundaries of converged domains with different mesh sizes in the bend optimization example of Section 2.5.1. Bottom : detail on the gray area featured in the top figure.*

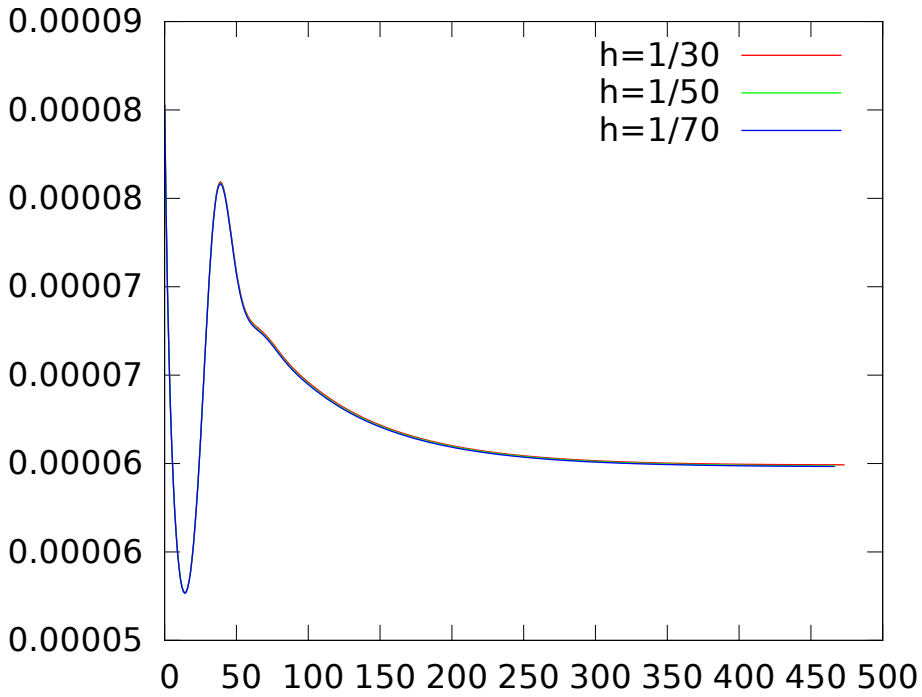


Figure 2.31 – *Evolution of the cost functional J for mesh sizes $h = 1/30$, $h = 1/50$ and $h = 1/70$.*

Figure 2.32 – *From top to bottom : Initial (left) and final (right) mesh for mesh sizes* $h = 1/30$, $h = 1/50$ *and* $h = 1/70$.

2.6 Conclusion and perspectives

In this chapter, we have presented a numerical framework for shape optimization in the context of fluid mechanics, consisting of well-established techniques which we have striven to present in an elementary and pedagogical way. The resulting strategy has been successfully applied to several benchmark test cases in the literature; admittedly, the techniques involved suffer from limitations, and there is a lot of room for improvements, notably:

- As we have explained in Section 2.4.4, the deformation of the computational mesh according to the shape gradient throughout the iterations of the optimization process is a delicate operation. Even though the heuristics described in Section 2.4.5 allow to overcome this difficulty in many cases, it may still happen that at some point the computational mesh becomes invalid; this is especially likely to happen when the evolving shape changes topology (for instance, two holes merge). This stake is a burning issue in the literature, and it calls for other means to represent shapes numerically than by a computational mesh, e.g. via the level set method [120, 44], or the SIMP method [96]. To keep a valid mesh, the gradient step must also be limited to small enough values, which can make convergence slow.

- The augmented Lagrangian algorithm described in Section 2.4.3 is well-tailored to impose one or two equality constraints on shapes. However, many natural constraints are inequality constraints, and it may be desirable to impose several of them. In such a case, it would be necessary to rely on more elaborated constrained optimization algorithms, such a Sequential Linear Programming (SLP); see for instance [111] about this point.

Chapter 3

Geometry optimization of an hourglass-shaped aquaporin

This chapter is based on the paper "Modeling and optimization of hourglass-shaped aquaporins", M. Bonnivard, F. Omnès, and Y. Privat, Mathematical Models and Methods in Applied Sciences, 0(0):1–36, 0, 2017 [29]

This chapter is organized as follows. We first introduce basic ideas behind the simulation and optimization of aquaporins 3.1. In Section 3.2.1, we introduce the flow model used for aquaporins and a few tools necessary for the analysis of this model. After proving the well-posedness of the fluid equation on a fixed domain, we introduce the shape optimization problem for the dissipated energy and prove the existence of a solution for a class of uniformly regular domains 3.3.

In the second part of the chapter, we introduce a gradient-descent method with specificities for the regularization of the shape gradient 3.2.2. Section 3.4 is dedicated to numerical questions. After a precise description of the method, we present and comment numerical results obtained assuming the domain is axially symmetric.

3.1 State of the art

3.1.1 Physical modelling of aquaporins

Aquaporins have been proven to be accurately modeled by Stokes equations and appropriate boundary conditions. In [26], authors investigate the "slippage" phenomena occurring at the vicinity of plane hydrophobic boundaries. In the simplest case of ideal hydrophobic interfaces where only the microscopic and macroscopic length scales are considered, theoretical and experimental arguments lead to using Navier slip boundary conditions at the surface of the cones of the aquaporin Γ_1 (see Figure 3.2)

$$[\sigma(\mathbf{u}, p)\mathbf{n} + \beta\mathbf{u}]_\tau = 0$$
$$\mathbf{u} \cdot \mathbf{n} = 0$$

This set of boundary conditions is chosen in order to model the flow within an aquaporin. This set of boundary conditions is an alternative to the classical "no-slip" boundary condition. Theoretical and experimental observations strongly suggest the use of partial-slip boundary conditions [73].

In the literature, the relevant variable is the slip length b defined as

$$b = \frac{\nu}{\beta}$$

Geometrically speaking, the slip length is the depth inside the solid where a linear extrapolation of the velocity profile vanishes assuming the profile is parallel to the boundary, see Figure 3.1.

The same model can be used for carbon nanotubes, for which the slip-length is about 100 nm [91]. The planar model could also be used on layers of graphene, for which the slip-length is about 60 nm.

3.1.2 Numerical simulation of fluid flow in an aquaporin

Coming into being at the beginning of the 1980s, Microfluidics is a domain dealing with the behavior and manipulation of fluids geometrically constrained to a small, usually sub-millimeter, scale

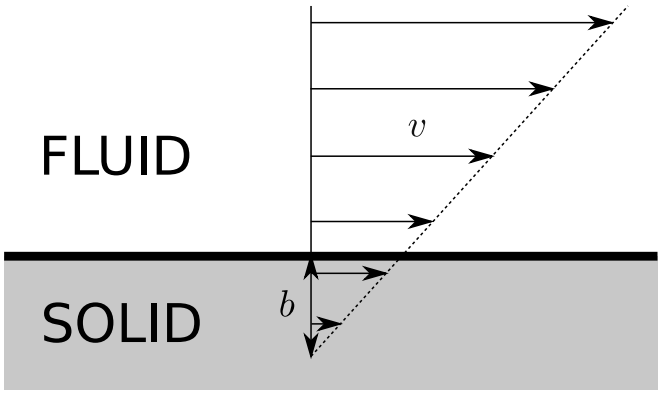


Figure 3.1 – Illustration of the slip-length b. The linear extrapolation of the velocity profile vanishes at depth b inside the solid.

at which capillary penetration governs the mass transport and its application can be widely found in DNA chips, lab-on-a-chip technology, micro-propulsion, micro-thermal technologies, etc [75]. Numerical simulations at small scale are thus required to study the specified physical and chemical properties of flows.

In particular, numerical simulation of membranes is expected to help imagining new efficient structures. Since the pioneering work of Hummer *et al.* [88], many simulations of fluid transport at a nanometric scale have been developed. These simulations can be classified into two categories:

- molecular dynamics (MD) models, that consider molecular interactions. See [140, 116] and [92, Chapter 16] for reference;
- methods based on continuum mechanics, that rely on the numerical resolution of the Navier-Stokes equations, using methods such as finite element methods (FEM) [74] or spectral elements methods [92, Chapter 14].

To the authors' best knowledge, MD has not yet been used to deal with shape-optimization problems. However, testing on a wide range of parameters, MD and FEM have been shown to provide highly similar results for hydrodynamic resistance [74], even though the continuum hypothesis does not to hold at a 0.4 nm scale, the estimated diameter the central channel in an aquaporin. In

particular, in this part of the aquaporin, measurements indicate that water molecules cross the central tube in a single file. However, previous experimental works also indicate that the Navier-Stokes equations hold when the channel diameter is more than ten times the size of a single fluid molecule [92, Chapter 10]. This reference also includes an example where the Navier-Stokes equations fail to describe the flow motion when this length ratio is only 4, in the case of a Poiseuille flow. For water, this critical diameter size is roughly 3 nm. As a partial conclusion, the Navier-Stokes equations provide an estimate of the velocity in the lateral cones, but not in the central tube. This is enough for our needs since we only aim to optimize the shape of the lateral cones.

Unlike MD simulations, FEM deal with continuous variables, making possible the use of the classical frameworks of fluid mechanics and shape-optimization. Note that optimizing molecular configurations seems challenging. Indeed, the computational cost associated to MD is usually high. Moreover, since only individual features of molecules or atoms are computed (speed, position), pressure and shear stress must be deduced in a second step. Yet, sensitivity analysis of energy functionals requires a good approximation of these quantities.

In [73], Gravelle *et al.* investigate an hourglass model for the aquaporin. From physical considerations, they impose partial-slip boundary conditions for the fluid on the channel walls. Varying the angle of the inlet and outlet cones, they show numerically that there exists an optimal angle minimizing the energy dissipation by the fluid inside the water channel. The class of admissible shapes is extended in [19], where the shape of the inlet cone is optimized through three design parameters. The optimal design parameters are found by trying numerous combinations, without using an optimization algorithm, as, for instance, a gradient method. While this approach is admittedly computationally costly, it revealed that there are no local minima for the hydrodynamic resistance in the three dimensional design space.

Using a similar model as in [73], we are interested in generalizing their work, by considering a wide family of admissible shapes, with the aim of analyzing more deeply the relationships between the aquaporin shape and its efficiency for permeating fluid. In this view, we will tackle the issue of minimizing the energy dissipated by the fluid through the channel constituting the structure, noting that this criterion is directly related to the performances of aquaporins. Concerning the modeling issues, we will assume that the fluid flux is known at the inlet, and that zero normal stress conditions (of Neumann type) are imposed at the outlet. It is notable that, in general, a flux condition does not allow to close a fluid model. Nevertheless, in our case, we will make a choice following the approach developed in [64] and derive a particular boundary condition implying the inlet flux condition, with the help of a dual variable.

3.2 Introduction and modeling of the problem

3.2.1 Geometry and fluid model

This section is devoted to modeling issues. To precise the framework of our study, we define the admissible geometries, as well as the fluid model, including our choices of boundary conditions.

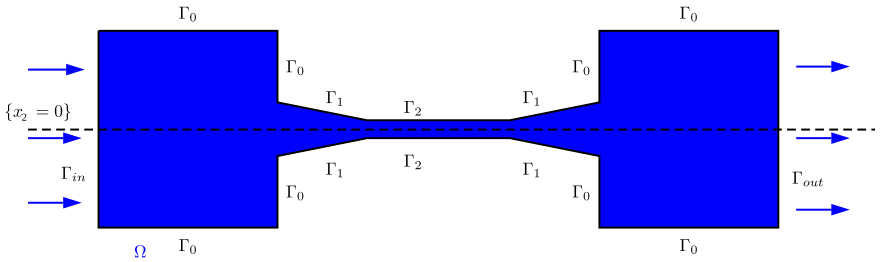


Figure 3.2 – *The domain Ω (two reservoirs connected by an hourglass shaped channel).*

In what follows, in order to deal with realistic shapes, we will consider connected and bounded domains Ω in $\mathbb{R}^2$. Domain Ω describes the geometry of the aquaporin. An example of such Ω is depicted on Fig. 3.2. We assume the domain Ω to be filled with

a viscous fluid of viscosity ν, with $\nu > 0$. The fluid domain Ω is made up of two reservoirs delimited by a lateral boundary Γ_0, and connected by a channel. The central part of the channel is tubular, its lateral boundary is denoted Γ_2; the inlet and outlet regions of the channel are conically shaped, with lateral boundary Γ_1. The upstream and downstream sections are labeled Γ_{in} and Γ_{out}, respectively.

Notation. We denote by $\mathbf{n}$ the outward unit normal vector to $\partial\Omega$, and for every smooth vector field φ defined on $\partial\Omega$, we define its tangential part by

$$\varphi_\tau := \varphi - (\varphi \cdot \mathbf{n})\mathbf{n}.$$

We define the strain tensor (symmetric part of the Jacobian matrix $\nabla\mathbf{u}$) by

$$e(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T),$$

as well as the stress tensor

$$\sigma(\mathbf{u}, p) = 2\nu e(\mathbf{u}) - p\mathrm{I}_2,$$

where $\mathbf{u}$ is the Eulerian velocity of the fluid, p is the pressure at every point $x \in \Omega$ and I_2 is the identity matrix in $\mathbb{R}^{2\times 2}$. We will denote by $\mathcal{H}^1$ the Hausdorff measure of dimension 1.

Fluid model and boundary conditions. The fluid motion is described by the Stokes equations

$$\begin{cases} -\operatorname{div}(\sigma(\mathbf{u}, p)) = 0 \quad \text{in } \Omega, & \text{(3.1a)} \\ \operatorname{div}\mathbf{u} = 0 \quad \text{in } \Omega. & \text{(3.1b)} \end{cases}$$

The momentum (3.1a) and mass-conservation (3.1b) equations are completed with boundary conditions. What follows is inspired by [27], where relevant boundary conditions on aquaporins are prescribed in order to obtain a closed physical fluid model, while prescribing the flux at the inlet of the considered structure.

- On the upstream section Γ_{in}, we assume that only the flow rate of the fluid is given. This condition reads

$$\int_{\Gamma_{in}} \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 = -Q, \qquad (3.2)$$

where the flow rate Q is a nonzero real number and $\mathbf{n}$ is the outward normal vector.

Recall that the average condition (3.2) is not sufficient to make the model well-posed and in particular to ensure the uniqueness of solutions. To overcome this difficulty, we follow the method developed in [64] to treat such "defective" boundary conditions involving averaged quantities instead of point-wise data on the boundary. Using this approach, condition (3.2) is interpreted as a (linear) constraint on the unknown $\mathbf{u}$ defined as a minimizer of an energy functional.

- In realistic applications, the conical regions of the channel are of nanometric size. At this scale, partial slip boundary conditions are considered relevant; they have been predicted theoretically and observed experimentally (see for instance [27]). Consequently, we set

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad [\sigma(\mathbf{u}, p)\mathbf{n}]_\tau + \beta \mathbf{u}_\tau = 0 \quad \text{on } \Gamma_1, \tag{3.3}$$

  where $\beta > 0$ is a constant friction parameter.

- Since the two reservoirs that are connected by the nanopore are typically of much larger size than the nanometric central channel, we impose the classical no-slip condition on their boundaries, that is,

$$\mathbf{u} = 0 \quad \text{on } \Gamma_0. \tag{3.4}$$

- On the lateral boundary Γ_2 of the central (tubular) part of the channel, we neglect the frictional dissipation by assuming perfect slip boundary conditions

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad [\sigma(\mathbf{u}, p)\mathbf{n}]_\tau = 0 \quad \text{on } \Gamma_2. \tag{3.5}$$

- Finally, on the downstream section Γ_{out} of the domain, we consider free outflow conditions, modeled by Neumann boundary conditions

$$\sigma(\mathbf{u}, p)\mathbf{n} = 0 \quad \text{on } \Gamma_{out}. \tag{3.6}$$

These boundary conditions are commonly used as passive conditions on artificial boundaries [85].

Let us introduce the functional space

$$V(\Omega) = \left\{\varphi \in H^1(\Omega, \mathbb{R}^2), \quad \varphi_{|\Gamma_0} = 0, \quad (\varphi \cdot \mathbf{n})_{|\Gamma_1 \cup \Gamma_2} = 0\right\}.$$

Due to the no-slip boundary condition imposed on Γ_0, the Poincaré inequality holds in $V(\Omega)$ and reads

$$\exists C > 0, \quad \forall \varphi \in V(\Omega) \quad \int_\Omega |\varphi|^2 \, \mathrm{d}x \leq C \int_\Omega |\nabla \varphi|^2 \, \mathrm{d}x,$$

where $|\cdot|$ stands either for the euclidean norm of a vector in $\mathbb{R}^2$, or a matrix in $\mathbb{R}^{2\times 2}$, depending on the context. As a result, $V(\Omega)$ is a Hilbert space for the inner product $V(\Omega)^2 \ni (\varphi_1, \varphi_2) \mapsto \int_\Omega \nabla \varphi_1 : \nabla \varphi_2 \, \mathrm{d}x$. We denote by $\|\cdot\|_{V(\Omega)}$ the norm associated with this inner product.

In the sequel, we will also need to use Korn inequality, whose validity in $V(\Omega)$ is another consequence of the no-slip boundary condition imposed on Γ_0 and the Lipschitz regularity of the boundary $\partial\Omega$. This inequality reads

$$\exists C_K > 0, \quad \forall \varphi \in V(\Omega) \quad \int_\Omega |\nabla \varphi|^2 \, \mathrm{d}x \leq C_K \int_\Omega |e(\varphi)|^2 \, \mathrm{d}x. \tag{3.7}$$

Mixed formulation of the Stokes problem with imposed inner flow through Γ_{in}.

For a given $Q \in \mathbb{R} \setminus \{0\}$, we consider the following problem: find $(\mathbf{u}_\lambda, p_\lambda) \in V(\Omega) \times L^2(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$\begin{cases} \forall \varphi \in V(\Omega) \quad 2\nu \int_\Omega e(\mathbf{u}_\lambda) : e(\varphi) \, \mathrm{d}x + \beta \int_{\Gamma_1} \mathbf{u}_\lambda \cdot \varphi \, \mathrm{d}\mathcal{H}^1 \\ \qquad - \int_\Omega p_\lambda \, \mathrm{div}\, \varphi \, \mathrm{d}x = \lambda \int_{\Gamma_{in}} \varphi \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 & \text{(3.8a)} \\ \forall q \in L^2(\Omega) \quad \int_\Omega q \, \mathrm{div}\, \mathbf{u}_\lambda \, \mathrm{d}x = 0 & \text{(3.8b)} \\ \int_{\Gamma_{in}} \mathbf{u}_\lambda \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 = -Q & \text{(3.8c)} \end{cases}$$

Remark 9. *The parameter λ appearing in (3.8a) can be regarded as the Lagrange multiplier associated to the constraint $\int_{\Gamma_{in}} \mathbf{u}_\lambda \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 =$*

$-Q$. Since both the constraint and the equations are linear, it will be made visible in the sequel that $-\lambda$ corresponds to the value of the normal constraint imposed on Γ_{in} to obtain the desired flow rate.

In view of showing the well-posed character of this variational equation, we state an "inf-sup" type lemma adapted to the definition of the space $V(\Omega)$.

Lemma 3. *The spaces $V(\Omega)$ and $L^2(\Omega)$ satisfy the inf-sup condition*

$$\inf_{q\in L^2(\Omega)\setminus\{0\}} \sup_{\varphi\in V(\Omega)\setminus\{0\}} \frac{\int_\Omega q \ \mathrm{div}\,\varphi \,\mathrm{d}x}{\|q\|_{L^2(\Omega)}\|\varphi\|_{V(\Omega)}} > 0. \tag{3.9}$$

Before showing Lemma 3, we state a useful preliminary result.

Lemma 4. *There exists a constant $C > 0$, depending only on Ω, such that for every $q \in L^2(\Omega)$, there exists $\mathbf{v} \in V(\Omega)$ satisfying*

$$\mathrm{div}\,\mathbf{v} = q \quad \textit{in } \Omega \quad \textit{and} \quad \|\mathbf{v}\|_{V(\Omega)} \le C\|q\|_{L^2(\Omega)}. \tag{3.10}$$

Proof. Let $q \in L^2(\Omega)$, and consider $\alpha_0 \in \mathcal{C}^\infty(\mathbb{R}^2)$, non identically null, with compact support, and such that $(\mathrm{spt}\,\alpha_0) \cap (\partial\Omega \setminus \Gamma_{in}) = \emptyset$ and $\int_{\Gamma_{in}} \alpha_0 \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 > 0$. For every $x \in \Omega$, we define

$$\alpha(x) := -\frac{\int_\Omega q(y)\,\mathrm{d}y}{\int_{\Gamma_{in}} \alpha_0 \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1}\ \alpha_0(x).$$

By construction, $\int_{\Gamma_{in}} \alpha\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = -\int_\Omega q(x)\,\mathrm{d}x$ and by Hölder inequality and the boundedness of α_0 and its derivatives, there exists a constant $C > 0$ such that $\|\alpha\|_{H^1(\Omega,\mathbb{R}^2)} \le C\,\|q\|_{L^2(\Omega,\mathbb{R}^2)}$. Moreover, $q - \mathrm{div}\,\alpha \in L^2(\Omega)$, and using Stokes formula and the properties of the support of α,

$$\int_\Omega (q - \mathrm{div}\,\alpha)\,\mathrm{d}x = \int_\Omega q\,\mathrm{d}x - \int_{\partial\Omega} \alpha\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = \int_\Omega q\,\mathrm{d}x + \int_{\Gamma_{in}} \alpha\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = 0.$$

Thus, since Ω is a Lipschitz domain, there exists a constant $C > 0$ depending only on Ω, and a vector field $\mathbf{v}_0 \in H^1_0(\Omega,\mathbb{R}^2)$ such that

$$\mathrm{div}\,\mathbf{v}_0 = q - \mathrm{div}\,\alpha \quad \text{and} \quad \|\mathbf{v}_0\|_{H^1_0(\Omega)} \le C\,\|q - \mathrm{div}\,\alpha\|_{L^2(\Omega)}$$

(see, for instance, [72] Corollary 2.4). Hence,

$$\|\mathbf{v}_0\|_{H^1(\Omega,\mathbb{R}^2)} \leq C\|q\|_{L^2(\Omega,\mathbb{R}^2)}.$$

Since $\alpha \in V(\Omega)$, the function $\mathbf{v}$ defined by $\mathbf{v} = \mathbf{v}_0 + \alpha$ belongs to $V(\Omega)$ and satisfies (3.10).

□

Let $q \in L^2(\Omega) \setminus \{0\}$ and $\mathbf{v} \in V(\Omega)$ such that (3.10) holds. Then

$$\sup_{\varphi \in V(\Omega)\setminus\{0\}} \frac{\int_\Omega q \ \mathrm{div}\, \varphi}{\|q\|_{L^2(\Omega)}\|\varphi\|_{V(\Omega)}} \geq \frac{\int_\Omega q \ \mathrm{div}\, \mathbf{v}}{\|q\|_{L^2(\Omega)}\|\mathbf{v}\|_{V(\Omega)}} = \frac{\|q\|_{L^2(\Omega)}}{\|\mathbf{v}\|_{V(\Omega)}} \geq C^{-1} > 0,$$

where C is defined in Lemma 4. Taking the infimum over $q \in L^2(\Omega) \setminus \{0\}$ yields the desired result.

The next proposition allows to interpret the solution $\mathbf{u}_\lambda$ of the Stokes system as a minimizer of an energy over a functional space.

Proposition 5. *For every $Q \in \mathbb{R} \setminus \{0\}$, there exists a unique triple $(\mathbf{u}_\lambda, p_\lambda, \lambda) \in V(\Omega) \times L^2(\Omega) \times \mathbb{R}$ satisfying (3.8a)-(3.8b)-(3.8c).*

Moreover, the function $\mathbf{u}_\lambda$ is the unique minimizer of the energy functional $\mathcal{E}_\Omega$ defined by

$$\mathcal{E}_\Omega(\mathbf{w}) = \nu \int_\Omega |e(\mathbf{w})|^2 \, \mathrm{d}x + \frac{\beta}{2} \int_{\Gamma_1} |\mathbf{w}|^2 \, \mathrm{d}\mathcal{H}^1 \tag{3.11}$$

over the space

$$V_{div}(\Omega) = V(\Omega) \cap \Big\{ \mathbf{w} \in H^1(\Omega,\mathbb{R}^2) \mid \mathrm{div}\, \mathbf{w} = 0 \ \textit{in}\ \Omega \\ \textit{and} \int_{\Gamma_{in}} \mathbf{w}\cdot\mathbf{n} \, \mathrm{d}\mathcal{H}^1 = -Q \Big\}. \tag{3.12}$$

This proof is an adaptation of the proof of [64, Proposition 2], to the case of a mix of no-slip, partial slip and perfect slip conditions on different parts of the boundary of the domain. For the sake of completeness, we recall it briefly.

Existence of a solution. Let $(\mathbf{u}_1, p_1) \in V(\Omega) \times L^2(\Omega)$ be the solution to the following (unconstrained) mixed formulation:

$$\begin{aligned} \forall \varphi \in V(\Omega) \quad & 2\nu \int_\Omega e(\mathbf{u}_1) : e(\varphi)\,\mathrm{d}x + \beta \int_{\Gamma_1} \mathbf{u}_1 \cdot \varphi\,\mathrm{d}\mathcal{H}^1 \\ & - \int_\Omega p_1\ \mathrm{div}\,\varphi\,\mathrm{d}x = \int_{\Gamma_{in}} \varphi \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 \qquad (3.13) \\ \forall q \in L^2(\Omega) \quad & \int_\Omega q\ \mathrm{div}\,\mathbf{u}_1\,\mathrm{d}x = 0 \qquad (3.14) \end{aligned}$$

By continuity of the trace operator $H^1(\Omega, \mathbb{R}^2) \to L^2(\Gamma_{in})$ and Hölder inequality, the linear operator $L : \varphi \in V(\Omega) \mapsto \int_{\Gamma_{in}} \varphi \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1$ is bounded. Hence, the existence and uniqueness of such $(\mathbf{u}_1, p_1) \in V(\Omega) \times L^2(\Omega)$ result from Korn inequality (3.7) and the inf-sup condition (3.9) (see [72], Lemma 4.1). Now, we set

$$\lambda = -\frac{Q}{\int_{\Gamma_{in}} \mathbf{u}_1 \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1}. \qquad (3.15)$$

By linearity of equations (3.13)–(3.14), defining $(\mathbf{u}_\lambda, p_\lambda) = \lambda(\mathbf{u}_1, p_1)$, we obtain a solution to system (3.8a)–(3.8c).

Uniqueness of the solution. Let $(\mathbf{y}, r, \lambda), (\mathbf{z}, s, \mu) \in V(\Omega) \times L^2(\Omega) \times \mathbb{R}$ be two solutions of equations (3.8a)–(3.8c). By linearity, we deduce from (3.8a) the relation

$$\begin{aligned} \forall \varphi \in V(\Omega) \quad & 2\nu \int_\Omega (e(\mathbf{y} - \mathbf{z})) : e(\varphi)\,\mathrm{d}x + \beta \int_{\Gamma_1} (\mathbf{y} - \mathbf{z}) \cdot \varphi\,\mathrm{d}\mathcal{H}^1 \\ & - \int_\Omega (r - s)\ \mathrm{div}\,\varphi\,\mathrm{d}x - (\lambda - \mu) \int_{\Gamma_{in}} \varphi \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 = 0. \qquad (3.16) \end{aligned}$$

Testing with $\varphi = \mathbf{y} - \mathbf{z}$ and using the relation $\int_{\Gamma_{in}} (\mathbf{y} - \mathbf{z}) \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 = 0$, we obtain

$$2\nu \int_\Omega |e(\mathbf{y} - \mathbf{z})|^2\,\mathrm{d}x + \beta \int_{\Gamma_1} |\mathbf{y} - \mathbf{z}|^2\,\mathrm{d}\mathcal{H}^1 = 0.$$

By Poincaré and Korn inequalities, this yields $\mathbf{y} = \mathbf{z}$ a.e. in Ω. Now, testing with $\varphi = \mathbf{y}$ in (3.16) and using the constraint $\int_{\Gamma_{in}} \mathbf{y} \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 = Q$, we deduce $(\lambda - \mu)\,Q = 0$, and so $\lambda = \mu$. Applying Lemma 4, there exists $\mathbf{w} \in V(\Omega)$ such that $\mathrm{div}\,\mathbf{w} = r - s$ a.e. in Ω. Thus, the relation $\int_\Omega (r - s)\ \mathrm{div}\,\mathbf{w}\,\mathrm{d}x = 0$ yields $\|r - s\|_{L^2(\Omega)} = 0$, which concludes the first part of the proof.

It remains to show that the function $\mathbf{u}_\lambda$ minimizes the energy functional $\mathcal{E}_\Omega$ over $V_{\mathrm{div}}(\Omega)$. To that end, let us consider any $\mathbf{v} \in V_{\mathrm{div}}(\Omega)$ and write $\mathbf{v} = \mathbf{u}_\lambda + \mathbf{h}$. Then, $\mathbf{h} \in V(\Omega)$ is divergence free and satisfies $\int_{\Gamma_{in}} \mathbf{h}\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = 0$. Using that $\mathbf{u}_\lambda$ satisfies (3.8a), one computes

$$\begin{aligned}\mathcal{E}_\Omega(\mathbf{v}) - \mathcal{E}_\Omega(\mathbf{u}_\lambda) &= \nu\int_\Omega |e(\mathbf{h})|^2\,\mathrm{d}x + \frac{\beta}{2}\int_{\Gamma_1}|\mathbf{h}|^2\,\mathrm{d}\mathcal{H}^1 - \lambda\int_{\Gamma_{in}}\mathbf{h}\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1\\ &= \nu\int_\Omega |e(\mathbf{h})|^2\,\mathrm{d}x + \frac{\beta}{2}\int_{\Gamma_1}|\mathbf{h}|^2\,\mathrm{d}\mathcal{H}^1 \geq 0.\end{aligned}$$

The expected conclusion follows.

Remark 10. *Notice that (3.8a)-(3.8b)-(3.8c) is the weak formulation of the partial differential equation*

$$\begin{cases} -\operatorname{div}(2\nu\, e(\mathbf{u}_\lambda)) + \nabla p_\lambda = 0 & \text{in } \Omega,\\ \operatorname{div}(\mathbf{u}_\lambda) = 0 & \text{in } \Omega,\\ \mathbf{u}_\lambda = 0 & \text{on } \Gamma_0,\\ \sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} + \lambda\mathbf{n} = 0 & \text{on } \Gamma_{in},\\ \sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} = 0 & \text{on } \Gamma_{out},\\ [\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n} + \beta\mathbf{u}_\lambda]_\tau = 0, \quad \mathbf{u}_\lambda\cdot\mathbf{n} = 0 & \text{on } \Gamma_1,\\ [\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}]_\tau = 0, \quad \mathbf{u}_\lambda\cdot\mathbf{n} = 0 & \text{on } \Gamma_2. \end{cases} \tag{3.17}$$

According to the proof of Proposition 5, we claim that

$$\lambda = -\frac{Q}{\int_{\Gamma_{in}}\mathbf{u}_1\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1}, \tag{3.18}$$

$(\mathbf{u}_1, p_1)$ being the solution of (3.8a)–(3.8b) with $\lambda = 1$.

This can be obtained by combining the two following facts: first, fixing $\lambda \in \mathbb{R}$, the system (3.17) has a unique weak solution (this is a byproduct of Proposition 5). Second, the mapping $\mathbb{R} \ni \lambda \mapsto (\mathbf{u}_\lambda, p_\lambda)$, where $(\mathbf{u}_\lambda, p_\lambda)$ denotes the unique weak solution of system (3.17), is linear.

Finally, we end this section by investigating the consequence of the symmetry assumptions on the domain Ω.

Mixed formulation of the Stokes problem with a symmetry condition.

In this paragraph, we adapt our model to the case where Ω is symmetric with respect to the axis $\{x_2 = 0\}$, that will be addressed numerically in Sec. 3.4. To this aim, we introduce some extra notation. We denote by $\mathcal{H}$ the hyperplane $\mathcal{H} = \{x_2 = 0\}$, and by $\mathrm{Ref}_{\mathcal{H}}$ the reflection through $\mathcal{H}$. We define $\mathcal{H}_+ = \{x_2 > 0\}$, $\Omega_+ = \Omega \cap \mathcal{H}_+$ the upper part of the domain, and $\Gamma_{sym} = \Omega \cap \mathcal{H}$ its lower boundary.

If Ω is symmetric with respect to $\mathcal{H}$, then regular solutions of Stokes problem (3.8a)-(3.8b)-(3.8c) enjoy nice symmetry properties, as stated in the following proposition.

Proposition 6. *Assuming that the solution $(\mathbf{u}, p)$ to the Stokes system (3.8a)-(3.8b)-(3.8c) belongs to $H^2(\Omega, \mathbb{R}^2) \times H^1(\Omega)$,*

$$\mathbf{u} = \mathrm{Ref}_{\mathcal{H}}(\mathbf{u} \circ \mathrm{Ref}_{\mathcal{H}}) \quad \textit{and} \quad p = p \circ \mathrm{Ref}_{\mathcal{H}} \quad \textit{a.e. in } \Omega, \tag{3.19}$$

and as a consequence,

$$[\sigma(\mathbf{u}, p)\mathbf{n}]_\tau = 0 \quad \textit{and} \quad \mathbf{u} \cdot \mathbf{n} = 0 \ \textit{on } \Gamma_{sym}. \tag{3.20}$$

where $\mathbf{n}$ is the outward-pointing normal vector on Γ_{sym}.

Let us use the following notation: for a vectorial function $\mathbf{y}$, we define $\tilde{\mathbf{y}} = \mathrm{Ref}_{\mathcal{H}}(\mathbf{y} \circ \mathrm{Ref}_{\mathcal{H}})$ and for a scalar function s, we define $\tilde{s} = s \circ \mathrm{Ref}_{\mathcal{H}}$. For all $(\varphi, q) \in V(\Omega) \times L^2(\Omega)$, one has

$$\begin{aligned} &2\nu \int_\Omega e(\tilde{\mathbf{u}}) : e(\varphi)\, \mathrm{d}x + \beta \int_{\Gamma_1} \tilde{\mathbf{u}} \cdot \varphi \, \mathrm{d}\mathcal{H}^1 - \int_\Omega \tilde{p}\ \mathrm{div}\ \varphi \, \mathrm{d}x \\ &= 2\nu \int_\Omega e(\mathbf{u}) : e(\tilde{\varphi})\, \mathrm{d}x + \beta \int_{\Gamma_1} \mathbf{u} \cdot \tilde{\varphi} \, \mathrm{d}\mathcal{H}^1 - \int_\Omega p\ \mathrm{div}\ \tilde{\varphi} \, \mathrm{d}x \\ &= \lambda \int_{\Gamma_{in}} \tilde{\varphi} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1, \end{aligned} \tag{3.21}$$

the second line being obtained by change of variables, using that $\Omega = \mathrm{Ref}_{\mathcal{H}}(\Omega)$ as well as the symmetry of boundary conditions. The third line follows from (3.8a)–(3.8c) on $\mathbf{u}$. From a similar change of variables and by symmetry of $\mathbf{n}$,

$$\begin{aligned} \int_{\Gamma_{in}} \tilde{\varphi} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 &= \int_{\Gamma_{in}} \tilde{\varphi} \cdot \tilde{\mathbf{n}} \, \mathrm{d}\mathcal{H}^1 = \int_{\Gamma_{in}} \widetilde{\varphi \cdot \mathbf{n}} \, \mathrm{d}\mathcal{H}^1 \\ &= \int_{\mathrm{Ref}_{\mathcal{H}}(\Gamma_{in})} \varphi \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1 = \int_{\Gamma_{in}} \varphi \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1. \end{aligned}$$

Using the same kind of arguments, $(\tilde{\mathbf{u}}, \tilde{p})$ also satisfies (3.8b)–(3.8c). By uniqueness of the solution of (3.8a)–(3.8c), we then claim that $(\tilde{\mathbf{u}}, \tilde{p}) = (\mathbf{u}, p)$. This shows the first claim of the proposition.

Now, since $u_2 \in H^1(\Omega)$ and according to the symmetry property we have just proved, one has $u_2 = -u_2$ a.e. on Γ_{sym}, and thus

$$\mathbf{u} \cdot \mathbf{n} = u_2 = 0 \text{ on } \Gamma_{sym}. \tag{3.22}$$

Denoting $f = \sigma(\mathbf{u}, p)\mathbf{e}_2$ and since $f \in H^1(\Omega, \mathbb{R}^2)$, one has,

$$\begin{aligned} f \circ \mathrm{Ref}_{\mathcal{H}} &= \begin{pmatrix} \nu\,(\partial_2 u_1 \circ \mathrm{Ref}_{\mathcal{H}} + \partial_1 u_2 \circ \mathrm{Ref}_{\mathcal{H}}) \\ 2\nu\,\partial_2 u_2 \circ \mathrm{Ref}_{\mathcal{H}} - p \circ \mathrm{Ref}_{\mathcal{H}} \end{pmatrix} \\ &= \begin{pmatrix} -\nu\,(\partial_2 u_1 + \partial_1 u_2) \\ 2\nu\,\partial_2 u_2 - p \end{pmatrix} = -\,\mathrm{Ref}_{\mathcal{H}} \circ f. \end{aligned}$$

Projecting this equality on axis $\mathbf{e}_1$, we obtain $f \cdot \mathbf{e}_1 = -f \cdot \mathbf{e}_1$ a.e. on Γ_{sym}. We then infer that

$$[\sigma(\mathbf{u}, p)\mathbf{n}]_\tau = 0 \text{ on } \Gamma_{sym}. \tag{3.23}$$

Hence, in the symmetric case, the flow is fully described by its restriction to the upper part Ω_+ of the domain. Besides, using the symmetry boundary condition on Γ_{sym} (3.20) leads to modifying the weak formulation (3.8a)-(3.8b)-(3.8c) as follows: for $Q \in \mathbb{R} \setminus \{0\}$, find $(\mathbf{u}_\lambda, p_\lambda) \in \tilde{V}(\Omega_+) \times L^2(\Omega_+)$ and $\lambda \in \mathbb{R}$ such that $\forall \varphi \in \tilde{V}(\Omega_+)$,

$$\begin{cases} 2\nu \displaystyle\int_{\Omega_+} e(\mathbf{u}_\lambda) : e(\varphi)\,\mathrm{d}x + \beta \int_{\Gamma_1 \cap \mathcal{H}_+} \mathbf{u}_\lambda \cdot \varphi\,\mathrm{d}\mathcal{H}^1 \\ \qquad - \displaystyle\int_{\Omega_+} p_\lambda\ \mathrm{div}\,\varphi\,\mathrm{d}x = \lambda \int_{\Gamma_{in} \cap \mathcal{H}_+} \varphi \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 & \text{(3.24a)} \\ \forall q \in L^2(\Omega_+) \quad \displaystyle\int_{\Omega_+} q\ \mathrm{div}\,\mathbf{u}_\lambda\,\mathrm{d}x = 0 & \text{(3.24b)} \\ \displaystyle\int_{\Gamma_{in} \cap \mathcal{H}_+} \mathbf{u}_\lambda \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 = -\frac{Q}{2} & \text{(3.24c)} \end{cases}$$

where

$$\tilde{V}(\Omega_+) := \left\{ \varphi \in H^1(\Omega_+, \mathbb{R}^2), \quad \varphi_{|\Gamma_0} = 0, \quad (\varphi \cdot \mathbf{n})_{|\Gamma_1 \cup \Gamma_2 \cup \Gamma_{sym}} = 0 \right\}.$$

All the considerations of the previous paragraph still hold true in that case, justifying the well-posed character of this formulation.

Moreover, Proposition 6 emphasizes that both formulations coincide when one assumes that Ω is symmetric w.r.t. the axis $\{x_2 = 0\}$, hence we can work with the simplified formulation (3.24a)-(3.24b)-(3.24c).

Dealing with symmetrical domains Ω will not only allow to integrate a kind of manufacturing constraint since it may appear difficult to design non-symmetric shapes, but also to simplify the problem.

3.2.2 The shape optimization problem

From a physical point of view, it is reasonable to look for a shape minimizing the energy dissipated by the fluid inside the aquaporin. Indeed, physically, this criterion accounts for the viscous effects responsible for the irreversible conversion of mechanical energy into internal energy or heat.

The cost functional is defined by

$$J(\Omega) = 2\nu \int_\Omega |e(\mathbf{u}_{\Omega,\lambda})|^2 \, \mathrm{d}x + \beta \int_{\Gamma_1} |\mathbf{u}_{\Omega,\lambda}|^2 \, \mathrm{d}\mathcal{H}^1, \tag{3.25}$$

where the triple $(\mathbf{u}_{\Omega,\lambda}, p_{\Omega,\lambda}, \lambda) \equiv (\mathbf{u}_\lambda, p_\lambda, \lambda) \in V(\Omega) \times L^2(\Omega) \times \mathbb{R}$ is defined in Proposition 5. Notice that $J(\Omega)$ also reads

$$J(\Omega) = 2 \min_{\mathbf{w} \in \tilde{V}_{\mathrm{div}}(\Omega)} \mathcal{E}_\Omega(\mathbf{w}),$$

where $\tilde{V}_{\mathrm{div}}(\Omega)$ and $\mathcal{E}_\Omega(\cdot)$ are defined in Proposition 5.

Since our main objective is to improve the results obtained in [73], we propose the following optimization strategy.

- First, we determine numerically the optimal angle of the inlet and outlet cones, minimizing the energy dissipation $J(\Omega)$ among all the domains Ω whose geometry is described by Figure 3.2. This step is a reproduction of the analysis performed in [73], and leads to obtaining a reference domain Ω^*, characterized by the position of the conical part Γ_1^* of its boundary (or equivalently, by the position of the terminal points of the lateral part Γ_0^*) .

- The improvement that we propose consists then in optimizing the shape of the conical regions of the domain, starting from the domain Ω^* that was determined in the previous step. In this process, the only part of the boundary that will be deformed is Γ_1^*; the rest of the boundary of Ω^* is fixed.

Let us give more details on the second step, that is, the shape optimization of boundary Γ^1, starting from the optimal cone of boundary Γ_1^*. We define the class of admissible shapes by

$$\begin{aligned}\mathcal{O}_{ad} = \{&\Omega \text{ open connected with} \\ &\text{a Lipschitz boundary, } \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_0^* \cup \Gamma_2 \subset \partial\Omega\}\,.\end{aligned} \tag{3.26}$$

The resulting shape optimization problem reads

$$\inf_{\Omega \in \mathcal{O}_{ad}} J(\Omega). \tag{3.27}$$

As previously stated, in the numerical simulations, we will restrict the admissible shapes to the ones that are symmetric with respect to the hyperplane $\mathcal{H}$. In that case, relying on Proposition 6, the cost functional can be expressed as

$$J(\Omega) = 2J_{sym}(\Omega_+) \quad \text{where} \quad J_{sym}(\Omega_+) = 2 \min_{\mathbf{w} \in \tilde{V}_{\text{div}}(\Omega_+)} \mathcal{E}_{\Omega_+}(\mathbf{w}), \tag{3.28}$$

and

$$\tilde{V}_{\text{div}}(\Omega_+) := \left\{\varphi \in \tilde{V}(\Omega_+), \quad \text{div}\,\varphi = 0 \ \text{ a.e on } \Omega^+\right\}.$$

3.3 Analysis of the shape optimization problem

This section is devoted first to the statement of an existence result for the shape optimization problem (3.27), and second, to the writing of the first order necessary optimality conditions for this problem.

3.3.1 Existence issues

It can be noted that the class $\mathcal{O}_{ad}$ defined by (3.26) is obviously not closed for usual domains topologies such as the Hausdorff complementary topology or the one associated to the strong L^1 convergence of

characteristic functions. To avoid the emergence of irregular shapes, for which the PDE model described in Section 3.2.1 makes no sense, we choose to impose geometrical constraints on the free boundary Γ_1, the varying part of the geometry of the admissible sets.

First, one wants to deal with (at least) Lipschitz domains since the definition of the functional space $V(\Omega)$ involves the outward pointing normal vector, and since such regularity is required for using standard tools in the analysis of variational problems in Fluids Mechanics, such as Korn inequality. However, a minimizing sequence of Lipschitz domains may converge to a very irregular domain; we refer for instance to [9, 81] for examples of such ill-posed optimization problems. Notice that recent works (see [37, 36]) have highlighted that when considering shape optimization problems involving the solution of an elliptic PDE with Robin boundary conditions, minimizing sequences of domains may become very irregular and lead to the emergence of inner cracks.

A satisfying framework to deal with Robin boundary conditions in shape optimization has been introduced in [37, 36]. It is based on a relaxation procedure, that consists in extending by 0 all test functions in the energy functional and embedding the free boundary problem into a larger class of functions, namely a subspace of special functions of bounded variation introduced originally by De Giorgi and Ambrosio. Unfortunately, adapting the approach of [37, 36] does not seem obvious. Indeed, this is due to

- the particular boundary conditions we consider, involving the normal and tangential parts of the vector field $\mathbf{u}$ and its derivative;
- the specificities of Fluids Mechanics equations, and in particular the divergence-free condition, which make it much more complicated to obtain a relaxed formulation of the PDE and the shape optimization problem (3.29). Notice also that the compactness theorems for SBV functions are not well adapted to dealing with symmetrized parts of gradients. In particular, it is not clear how to adapt the Korn inequality when considering

domains with a boundary that is not Lipschitz regular.

A possible solution consists in restricting the class of admissible domains, by assuming some kind of uniform Lipschitz regularity. For that purpose, let us define the notion of ε-cone property, introduced in [46].

Definition 2. *Let y be a point of $\mathbb{R}^2$, ξ a normalized vector and $\varepsilon > 0$. We denote by $C(y, \xi, \varepsilon)$, the unpointed cone*

$$C(y,\xi,\varepsilon) = \{z \in \mathbb{R}^2, \langle z-y, \xi\rangle \geq \cos\varepsilon \|z-y\| \text{ and } 0 < \|z-y\| < \varepsilon\}.$$

We say that an open set Ω verifies the ε-cone property if

$$\forall x \in \partial\Omega, \exists \xi_x \in \mathbb{S}^1, \forall y \in \overline{\Omega} \cap B(x,\varepsilon), C(y,\xi_x,\varepsilon) \subset \Omega.$$

Another geometrical constraint, which is standard in shape optimization, is to assume that all admissible shapes are contained in a compact set D to avoid the degeneracy of the free boundary. For this reason, let us introduce an external box D, defined as the convex hull of Ω; in other words, D is the rectangle of sides $\Gamma_{in}, \Gamma_{out}$, and whose orthogonal sides contain the segments in Γ_0 that are parallel to the flow [1] (see Fig. 3.3).

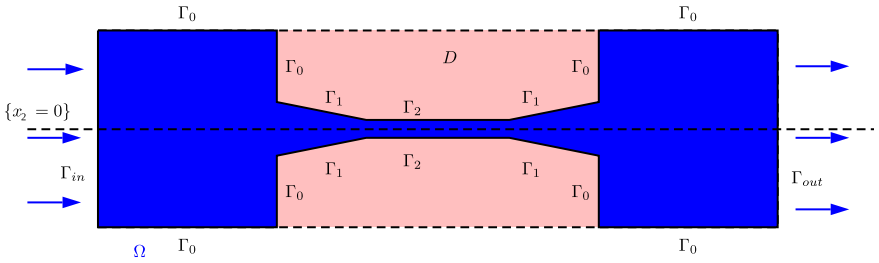


Figure 3.3 – *The external, rectangular box D contains the domain Ω (in blue), completed with two lateral sub-regions (in pink).*

Finally, the shape optimization problem that we investigate reads

$$\boxed{\inf\{J(\Omega),\ \Omega \in \mathcal{O}_{ad},\ \Omega \subset D \text{ and } \Omega \text{ satisfies the } \varepsilon\text{-cone property}\}}, \tag{3.29}$$

[1] Obviously, any other choice of compact set would be convenient from the point of view of existence theory. This particular choice is motivated by the biological/technological applications described in Section 1.2.3.

for some given parameter $\varepsilon > 0$.

One has the following existence result.

Theorem 7. *The shape optimization problem (3.29) has a solution.*

Let us first recall some convergence and topological notions for the elements of $\mathcal{O}_{\text{ad}}$.

Definition 3. *Recall that D denotes a given compact set. A sequence of open domains $(\Omega_n)_{n\geq 0}$ compactly embedded in D is said*

- *converging to Ω for the Hausdorff convergence if*

$$\lim_{n\to+\infty} d_{\mathcal{H}}(D\backslash\Omega_n, D\backslash\Omega) = 0,$$

  where $d_{\mathcal{H}}(K_1, K_2) = \max(\rho(K_1, K_2), \rho(K_2, K_1))$, for any $(i, j) \in \{1, 2\}^2$, $\rho(K_i, K_j) = \sup_{x\in K_i} d(x, K_j)$, and $\forall x \in D$, $d(x, K_i) = \inf_{y\in K_i} d(x, y)$;

- *converging to Ω in the sense of characteristic functions if for all $p \in [1, +\infty)$,*

$$\chi_{\Omega_n} \xrightarrow[n\to\infty]{} \chi_\Omega \text{ in } L^p_{loc}(\mathbb{R}^2);$$

- *converging to Ω in the sense of compacts if*

  1. $\forall K$ compact subset of $D, K \subset \Omega \Rightarrow \exists n_0 \in \mathbb{N}^*, \ \forall n \geq n_0, \ K \subset \Omega_n$;
  2. $\forall K$ compact subset of $D, K \subset D\backslash\overline{\Omega} \Rightarrow \exists n_0 \in \mathbb{N}^*, \ \forall n \geq n_0, \ K \subset D\backslash\overline{\Omega_n}$.

We first stress that the class of admissible domains is closed and compact at the same time for the Hausdorff topology, the convergence of characteristic functions and in the sense of compacts. Indeed, this is a direct consequence of stability with respect to inclusion for the Hausdorff topology, as well as the closure of the set of domains satisfying the ε-cone condition for the three aforementioned topologies.

Let $(\Omega_n)_{n\in\mathbb{N}}$ be a minimizing sequence for Problem (3.29). Since the open sets Ω_n are contained in a fixed compact set D, there exists

a subsequence, still denoted (with a slight abuse of notation) by Ω_n converging (for the Hausdorff distance, but also for the other usual topologies) to some set Ω. Moreover, according to the remark above about the closure of admissible sets, Ω belongs to the class $\mathcal{O}_{ad}$, $\Omega \subset D$ and Ω satisfies the ε-cone property (see e.g. [81, Theorem 2.4.10]).

To prove the existence result, it remains to show the lower-semicontinuity of the criterion J. For every $n \in \mathbb{N}$, we denote by $(\mathbf{u_n}, p_n, \lambda_n) \in V(\Omega_n) \times L^2(\Omega_n) \times \mathbb{R}$ the unique triple satisfying (3.8a)-(3.8b)-(3.8c) (see Proposition 5).

Since $(\Omega_n)_{n\in\mathbb{N}}$ is a minimizing sequence for Problem (3.29), we infer that the sequence

$$\left(\max\left\{\int_{\Omega_n} |e(\mathbf{u}_n)|^2 \,\mathrm{d}x, \int_{\Gamma_1^n} |\mathbf{u}_n|^2 \,\mathrm{d}\mathcal{H}^1\right\}\right)_{n\in\mathbb{N}},$$

with $\Gamma_1^n = \partial\Omega_n \backslash (\Gamma_{in} \cup \Gamma_0 \cup \Gamma_2 \cup \Gamma_{out})$, is bounded.

This shows that, up to subsequences, $(\mathbf{u_n}\mathbb{1}_{\Omega_n})$ converges weakly to some function $\mathbf{v} \in L^2(D, \mathbb{R}^2)$ whereas $(\mathbf{D}(\mathbf{u_n})\mathbb{1}_{\Omega_n})$ converges weakly to some function $\mathbf{z} \in L^2(D, \mathcal{S}_2(\mathbb{R}))$ (the notation $\mathcal{S}_2(\mathbb{R})$ denoting the set of real-symmetric matrices of $\mathbb{R}^2$).

Let us show the existence of $\mathbf{u} \in H^1(\Omega, \mathbb{R}^2)$ such that $\mathbf{v} = \mathbf{u}\mathbb{1}_\Omega$ and $\mathbf{z} = \mathbf{D}(\mathbf{u})\mathbb{1}_\Omega$. For every $\varphi \in L^2(D, \mathbb{R}^2)$, one has

$$\int_D \mathbf{u_n}\mathbb{1}_{\Omega_n} \cdot \varphi \,\mathrm{d}x \xrightarrow[n\to+\infty]{} \int_D \mathbf{v}\mathbb{1}_\Omega \cdot \varphi \,\mathrm{d}x = \int_D \mathbf{v} \cdot \varphi \,\mathrm{d}x,$$

and therefore, $\mathbf{v} = \mathbf{v}\mathbb{1}_\Omega$. Similarly, for every $\psi \in L^2(D, \mathcal{M}_d(\mathbb{R}))$, one has

$$\int_D \mathbf{D}(\mathbf{u_n})\mathbb{1}_{\Omega_n} : \varphi \,\mathrm{d}x \xrightarrow[n\to+\infty]{} \int_D \mathbf{z}\mathbb{1}_\Omega \cdot \varphi \,\mathrm{d}x = \int_D \mathbf{z} : \psi \,\mathrm{d}x,$$

so that $\mathbf{z} = \mathbf{z}\mathbb{1}_\Omega$. Let $\mathbf{u}$ be the restriction of $\mathbf{v}$ to Ω, one has for all $(i,j) \in \{1,2\}^2$ and $\varphi \in C_c^\infty(\overline{\Omega})$,

$$\begin{aligned}\lim_{n\to+\infty} \int_D \mathbb{1}_{\Omega_n} u_{n,i} \frac{\partial\varphi}{\partial x_j} \,\mathrm{d}x &= \int_D \mathbb{1}_\Omega u_i \frac{\partial\varphi}{\partial x_j} \,\mathrm{d}x \\ &= -\lim_{n\to+\infty} \int_D \mathbb{1}_{\Omega_n} \varphi \frac{\partial u_{n,i}}{\partial x_j} \,\mathrm{d}x = -\int_D \mathbb{1}_\Omega \varphi z_{ij} \,\mathrm{d}x\end{aligned}$$

by using the Green formula and that $\varphi \in C_c^\infty(\overline{\Omega_n})$ for n large enough. As a consequence, there holds $\mathbf{z} = \nabla \mathbf{u} \mathbb{1}_\Omega$, showing the weak convergence of $(\mathbf{D}(\mathbf{u_n})\mathbb{1}_{\Omega_n})_{n\in\mathbb{N}}$ to $\mathbf{D}(\mathbf{u})\mathbb{1}_\Omega$.

To prove the strong convergence of $(\mathbf{u_n}\mathbb{1}_{\Omega_n})_{n\in\mathbb{N}}$ in $L^2(D,\mathbb{R}^2)$ to $\mathbf{u}\mathbb{1}_\Omega$ as well as the semicontinuity of the boundary term of J, we will adapt [37, Lemma 4.2].

This way, we only underline the slight changes needed to get the result. The first step consists in investigating the behavior of the minimizing sequence on sub-domains and contains the main differences with [37, Lemma 4.2]. Hence, let us consider a subset $\hat{\Omega}$ having a compact closure in Ω and a Lipschitz boundary. Using the convergence in the sense of compacts, we know that $\hat{\Omega} \subset \Omega_n \subset \Omega$ for n large enough. Therefore, the function $\mathbf{u_n}$ belongs to $H^1(\hat{\Omega},\mathbb{R}^2)$ for n large enough. By using the Rellich-Kondratov embedding theorem, one infers that $(\mathbf{u_n})_{n\in\mathbb{N}}$ converges strongly in $L^2(\hat{\Omega},\mathbb{R}^2)$ and weakly in $H^1(\hat{\Omega},\mathbb{R}^2)$ to $\mathbf{u}$. This follows in particular from the Korn inequality in $\hat{\Omega}$ which asserts that the usual H^1-norm is equivalent to the norm $\|\cdot\|_{L^2} + \|\mathbf{D}(\cdot)\|_{L^2}$, since there exists a subset of $\partial\Omega$ of positive Hausdorff measure (namely Γ_0) on which homogeneous Dirichlet boundary conditions are imposed.

To conclude, it remains to investigate the global behavior of the sequence on the whole domain. By using the strong convergence on sub-domains $\hat{\Omega}$ we have just proved, the proof consists of a direct adaptation of [37, Proof of Lemma 4.2]. For this reason, we refer to [37, Proof of Lemma 4.2] and infer successively that $(\mathbf{u_n}\mathbb{1}_{\Omega_n})_{n\in\mathbb{N}}$ converges strongly to $\mathbf{u}\mathbb{1}_\Omega$ in $L^2(D,\mathbb{R}^2)$ and

$$\int_{\Gamma_1} |\mathbf{u}|^2 \mathrm{d}\mathcal{H}^1 \leq \liminf_{n\to+\infty} \int_{\Gamma_1^n} |\mathbf{u_n}|^2 \mathrm{d}\mathcal{H}^1,$$

with $\Gamma_1^n = \partial\Omega_n \backslash (\Gamma_{in} \cup \Gamma_0 \cup \Gamma_2 \cup \Gamma_{out})$ and $\Gamma_1 = \partial\Omega \backslash (\Gamma_{in} \cup \Gamma_0 \cup \Gamma_2 \cup \Gamma_{out})$.

Combining the previous results, we then infer that (up to subsequences)

$$\mathcal{E}_\Omega(\mathbf{u}) \leq \liminf_{n\to+\infty} J(\Omega_n).$$

To conclude, it remains to show that $\mathbf{u}$ belongs to the space $V_{\mathrm{div}}(\Omega)$ (defined by (3.12)). Notice first that, on fixed boundaries,

one has obviously $\mathbf{u}_{|\Gamma_0} = 0$, $\mathbf{u}_{|\Gamma_2} = 0$ and $\int_{\Gamma_{in}} \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = -Q$ (the last equality following from the weak H^1-convergence of the sequence $(\mathbf{u_n})_{n\in\mathbb{N}}$ in a neighborhood of Γ_{in} in Ω combined with the trace continuity property on Γ_{in}). It remains to show that

$$\operatorname{div}\mathbf{u} = 0 \quad \text{in } \Omega \qquad \text{and} \quad \mathbf{u}\cdot\mathbf{n} = 0 \quad \text{on } \Gamma_1.$$

Using an integration by parts, these two conditions can be gathered under the weak form

$$\int_\Omega \mathbf{u}\cdot\nabla\varphi\,\mathrm{d}x = 0,$$

for every test function $\varphi \in H^1(D)$ such that $\varphi = 0$ on $\Gamma_{in}\cup\Gamma_0\cup\Gamma_2\cup\Gamma_{out}$. This is obtained by passing to the limit in the equality

$$\int_D \mathbb{1}_{\Omega_n}\mathbf{u_n}\cdot\nabla\varphi\,\mathrm{d}x = 0,$$

where $\mathbf{u_n}$ has been extended by 0 to the whole compact set D and φ denotes any test function in $H^1(D)$ such that $\varphi = 0$ on $\Gamma_{in}\cup\Gamma_0\cup\Gamma_2\cup\Gamma_{out}$.

Note that a similar existence result was obtained for a shape optimization problem arising in Fluid Mechanics with homogeneous Dirichlet conditions on the free boundary in [82].

3.3.2 Computation of the shape derivative of J

We are interested in the differentiability of the solution $\mathbf{u}_\lambda \in V(\Omega)$ to system (3.8a)-(3.8b)-(3.8c), with respect to deformations of the domain Ω preserving $\Gamma_{in}, \Gamma_{out}, \Gamma_0$ and Γ_2, but acting on the shape of the "conical" boundary Γ_1. Let $\mathbf{V} \in W^{2,\infty}(\mathbb{R}^2,\mathbb{R}^2)$, with compact support, and such that $\mathbf{V}(x) = 0$ if $x \in \partial\Omega\setminus\Gamma_1$. Let us stress that such $W^{2,\infty}$ regularity of the deformation field is specific to the treatment of a slip boundary condition in a stationary model of Newtonian flow. Indeed, in the case of Dirichlet boundary conditions, it is enough to consider Lipschitz deformations (see for instance [20]). In the present case, the recasting of the variational formulation of the problem in the reference domain requires to preserve the non

penetration boundary condition on the the deformed boundary. This can be done by multiplying the velocity field by the Jacobian matrix of $\mathbf{V}$, to take into account the change of direction of the normal vector field. As a result, the energy associated with the system involves second derivatives of the displacement field, which explains the required smoothness on $\mathbf{V}$.

We introduce $T > 0$ and a mapping

$$t \in (-T, T) \mapsto \Phi_t = (\Phi_t^1, \Phi_t^2) \in W^{2,\infty}(\mathbb{R}^2, \mathbb{R}^2),$$

of class $\mathcal{C}^3$, satisfying the properties

$$\Phi_0 = \mathrm{Id}, \quad \frac{d\Phi_t}{dt}_{|t=0} = \mathbf{V}.$$

Moreover, we assume that $\Phi_t(x) = x$ for every $x \in \partial\Omega \setminus \Gamma_1$ and every $t \in (-T, T)$. A typical choice is given by $\Phi_t = \mathrm{Id} + t\mathbf{V}$.

We may choose T small enough so that for $t \in (-T, T)$, Φ_t is one to one and onto, and for every $x \in \mathbb{R}^2$, the mapping $t \in (-T, T) \mapsto \Phi_t^{-1}(x)$ is differentiable at $t = 0$, with

$$\frac{d}{dt}\left[\Phi_t^{-1}(x)\right]_{|t=0} = -\mathbf{V}(x).$$

For every $t \in (-T, T)$, we define $\Omega_t := \Phi_t(\Omega)$ and denote by $\mathbf{u}_{\lambda,t} \in V(\Omega_t)$ the solution of system (3.8a)-(3.8b)-(3.8c) for $\Omega = \Omega_t$.

Proposition 8. *Let $\Omega \in \mathcal{O}_{ad}$. The mapping*

$$t \in (-T, T) \mapsto (\mathbf{u}_{\lambda,t} \circ \Phi_t, p_{\lambda,t} \circ \Phi_t) \in H^1(\Omega) \times L^2(\Omega)$$

is differentiable at $t = 0$.

First step: differentiability of J and $\mathbf{u}$ with respect to the domain. Differentiability of volumic criteria with respect to domain variations is generally proved by using standard techniques resting upon the implicit function theorem (see e.g. [20] and [81, Theorems 5.3.1 and 5.3.2]). In the case that we investigate, we have to take into account particular boundary conditions, namely the slip boundary

conditions, which requires several adaptations. For this reason, we prove the shape-differentiability of J in detail. According to the proof of Proposition 5, the solution $\mathbf{u}_\lambda$ to problem (3.8a)–(3.8c) reads $\mathbf{u}_\lambda = \lambda\,\mathbf{u}_1$, where $\mathbf{u}_1$ is the solution to (3.13)-(3.14) and λ is given by (3.15). Consequently, it is enough to prove the differentiability of $\mathbf{u}_1$ with respect to domain variations to conclude that λ and $\mathbf{u}_\lambda$ are differentiable as well.

Let $(\mathbf{u}_{1,t}, p_{1,t}) \in V(\Omega_t) \times L^2(\Omega_t)$ be the unique solution to

$$\forall \varphi \in V(\Omega_t) \quad 2\nu \int_{\Omega_t} e(\mathbf{u}_{1,t}) : e(\varphi)\,\mathrm{d}x + \beta \int_{\Phi_t(\Gamma_1)} \mathbf{u}_{1,t}\cdot\varphi\,\mathrm{d}\mathcal{H}^1 - \int_{\Omega_t} p_{1,t}\,\operatorname{div}\varphi\,\mathrm{d}x = \int_{\Gamma_{in}} \varphi\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1, \tag{3.30}$$

$$\operatorname{div}\mathbf{u}_{1,t} = 0 \quad \text{in } \Omega_t. \tag{3.31}$$

To address the differentiability of $(\mathbf{u}_{1,t}, p_{1,t})$ with respect to t, we need to recast problem (3.30)-(3.31) into the reference domain Ω, by introducing a change of function performed on both the solution $(\mathbf{u}_{1,t}, p_{1,t})$ and the test functions (φ, q).

Let $t \in (-T, T)$ be fixed. We denote by $J_t(y) = D_y\Phi_t(y)$ the Jacobian matrix of Φ_t at point y, by $\mathcal{J}_t(y)$ its determinant and we define $M_t(y) = J_t(y)^{-1}$. We introduce the function $R_t \in L^2(\Omega)$ defined by $R_t = \mathcal{J}_t\, p_{1,t}\circ\Phi_t$. Since $p_{1,t} \in L^2(\Omega_t)$, using the change of variable associated with Φ_t, we see that $R_t \in L^2(\Omega)$. Now, consider the function $\mathbf{U}_t \in H^1(\Omega)$ defined by

$$\mathbf{U}_t = M_t\,(\mathbf{u}_{1,t}\circ\Phi_t).$$

The product with the matrix M_t is justified by the fact that the new function $\mathbf{U}_t$ satisfies the non penetration condition $\mathbf{U}_t\cdot\mathbf{n} = 0$ on Γ_1 (see, for instance, [28]). Consequently, by construction of the diffeomorphism Φ_t, $\mathbf{U}_t \in V(\Omega_t)$. The Jacobian matrix $\nabla\mathbf{u}_{1,t}$ and the divergence $\operatorname{div}\mathbf{u}_{1,t}$ are transformed as follows:

$$\nabla\mathbf{u}_{1,t}\circ\Phi_t = (H_t\mathbf{U}_t + J_t\,\nabla\mathbf{U}_t)\,M_t, \quad (\operatorname{div}\mathbf{u}_{1,t})\circ\Phi_t = \operatorname{Tr}\left[(H_t\mathbf{U}_t + J_t\,\nabla\mathbf{U}_t)\,M_t\right]$$

where $\operatorname{Tr}(A)$ is the trace of a square matrix A, and for any $\mathbf{U} \in V(\Omega)$, $H_t\mathbf{U} \in H^1(\Omega, \mathbb{R}^{2\times 2})$ is defined component by component by $(H_t\mathbf{U})_{i,j} = \sum_{k=1}^d \frac{\partial^2\Phi_i(t)}{\partial y_j \partial y_k}\,U_k$.

For every $\varphi \in V(\Omega_t)$, following the previous remarks, we can define $\tilde{\varphi} \in V(\Omega)$ by $\tilde{\varphi} = M_t\, \varphi \circ \Phi_t$. Problem (3.30)-(3.31) is then equivalent to

$$\begin{aligned}\forall \tilde{\varphi} \in V(\Omega) \qquad & \frac{\nu}{2} \int_\Omega \Big((H_t \mathbf{U}_t + J_t\, \nabla \mathbf{U}_t)\, M_t + [(H_t \mathbf{U}_t + J_t\, \nabla \mathbf{U}_t)\, M_t]^T \Big) : \\ & \Big((H_t \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t + [(H_t \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t]^T \Big)\; J_t \, \mathrm{d}y \\ & + \beta \int_{\Gamma_1} (J_t \mathbf{U}_t) \cdot (J_t \tilde{\varphi})\; \left| M_t^T \mathbf{n} \right|\; \mathcal{J}_t \, \mathrm{d}\mathcal{H}^1 - \int_\Omega R_t \, \mathrm{Tr}\left[(H_t\, \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t \right] \mathrm{d}y \\ & = \int_{\Gamma_{in}} \tilde{\varphi} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1, \end{aligned} \tag{3.32}$$

$$\mathrm{Tr}\left[(H_t\, \mathbf{U}_t + J_t\, \nabla \mathbf{U}_t)\, M_t \right] = 0 \quad \text{in } \Omega. \tag{3.33}$$

In view of equations (3.32)-(3.33), we introduce the operator

$$\begin{aligned} F : (-T, T) \times V(\Omega) \times L^2(\Omega) &\to [V(\Omega)]' \times L^2(\Omega) \\ (t, \mathbf{U}, R) &\mapsto (F_1(t, \mathbf{U}, R), F_2(t, \mathbf{U}, R)) \end{aligned}$$

where for all $\tilde{\varphi} \in V(\Omega)$, one has

$$\begin{aligned} & \langle F_1(t, \mathbf{U}, R), \tilde{\varphi} \rangle_{[V(\Omega)]' \times V(\Omega)} = \\ & \frac{\nu}{2} \int_\Omega \Big((H_t \mathbf{U} + J_t\, \nabla \mathbf{U})\, M_t + [(H_t \mathbf{U} + J_t\, \nabla \mathbf{U})\, M_t]^T \Big) : \\ & \Big((H_t \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t + [(H_t \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t]^T \Big)\; J_t \, \mathrm{d}y \\ & + \beta \int_{\Gamma_1} (J_t \mathbf{U}) \cdot (J_t \tilde{\varphi})\; \left| M_t^T \mathbf{n} \right|\; \mathcal{J}_t \, \mathrm{d}\mathcal{H}^1 \\ & - \int_\Omega R \, \mathrm{Tr}\left[(H_t\, \tilde{\varphi} + J_t\, \nabla \tilde{\varphi})\, M_t \right] \mathrm{d}y - \int_{\Gamma_{in}} \tilde{\varphi} \cdot \mathbf{n} \; \mathrm{d}\mathcal{H}^1, \\ & F_2(t, \mathbf{U}, R) = \mathrm{Tr}\left[(H_t\, \mathbf{U} + J_t\, \nabla \mathbf{U})\, M_t \right]. \end{aligned}$$

For every $t \in (-T, T)$, by uniqueness of the solution $(\mathbf{u}_{1,t}, p_{1,t}) \in V(\Omega_t) \times L^2(\Omega_t)$ to (3.30)-(3.31), there exists a unique pair $(\mathbf{U}_t, R_t) \in V(\Omega) \times L^2(\Omega)$ such that $F(t, \mathbf{U}_t, R_t) = 0$. We will apply the implicit function theorem to prove that the mapping $t \mapsto (\mathbf{U}_t, R_t)$ is differentiable at $t = 0$. Since the mapping $t \in (-T, T) \mapsto \Phi_t \in W^{2,\infty}(\mathbb{R}^2, \mathbb{R}^2)$ is of class $\mathcal{C}^1$, every coefficient appearing in the operator H_t and the matrices J_t, M_t is of class $\mathcal{C}^1$ in t. Consequently,

F is of class $\mathcal{C}^1$ with respect to $(t, \mathbf{U}, R)$, and its differential with respect to $(\mathbf{U}, R)$ at point $(0, \mathbf{U}_0, R_0)$ reads

$$\begin{aligned}
&\forall (\mathbf{Z}, S) \in V(\Omega) \times L^2(\Omega) \quad \forall \varphi \in V(\Omega) \\
&\left\langle D_{(\mathbf{U},R)} F_1(0, \mathbf{U}_0, R_0)(\mathbf{Z}, S), \varphi \right\rangle = 2\nu \int_\Omega e(\mathbf{Z}) : e(\varphi)\, \mathrm{d}y \\
&\qquad + \beta \int_{\Gamma_1} \mathbf{Z} \cdot \varphi \, \mathrm{d}\mathcal{H}^1 - \int_\Omega S \ \mathrm{div}\, \varphi \, \mathrm{d}y, \\
&D_{(\mathbf{U},R)} F_2(0, \mathbf{U}_0, R_0)(\mathbf{Z}, S) = \mathrm{div}\, \mathbf{Z}.
\end{aligned}$$

Let us prove that $D_{(\mathbf{U},R)} F(0, \mathbf{U}_0, R_0) \in \mathcal{L}(V(\Omega) \times L^2(\Omega), [V(\Omega)]' \times L^2(\Omega))$ is an isomorphism. To this end, consider $(\mathbf{G}, s) \in [V(\Omega)]' \times L^2(\Omega)$. Since $V(\Omega)$ is a Hilbert space for the scalar product $(\mathbf{U}, \varphi) \in V(\Omega) \times V(\Omega) \mapsto \int_\Omega \nabla \mathbf{U} : \nabla \varphi \, \mathrm{d}x$, by Riesz theorem we can identify $\mathbf{G}$ with its representative in $V(\Omega)$, and define for every $\varphi \in V(\Omega)$ the duality pairing

$$\langle \mathbf{G}, \varphi \rangle_{[V(\Omega)]', V(\Omega)} := \int_\Omega \nabla \mathbf{G} : \nabla \varphi \, \mathrm{d}x.$$

By lemma 4, there exists a constant $C > 0$ and a function $\mathbf{v} \in V(\Omega)$ such that $\mathrm{div}\, \mathbf{v} = s$ a.e. in Ω and $\|\mathbf{v}\|_{V(\Omega)} \leq C \|s\|_{L^2(\Omega)}$. Now, define $(\mathbf{Z}_0, S) \in V(\Omega) \times L^2(\Omega)$ as the unique solution to the following problem:

$$\begin{aligned}
\forall \varphi \in V(\Omega) \qquad & 2\nu \int_\Omega e(\mathbf{Z}_0) : e(\varphi)\, \mathrm{d}y + \beta \int_{\Gamma_1} \mathbf{Z}_0 \cdot \varphi \, \mathrm{d}\mathcal{H}^1 - \int_\Omega S \ \mathrm{div}\, \varphi \, \mathrm{d}y \\
&= \int_\Omega \nabla \mathbf{G} : \nabla \varphi \, \mathrm{d}x - 2\nu \int_\Omega e(\mathbf{v}) : e(\varphi)\, \mathrm{d}y - \beta \int_{\Gamma_1} \mathbf{v} \cdot \varphi \, \mathrm{d}\mathcal{H}^1, \\
& \qquad\qquad \mathrm{div}\, \mathbf{Z}_0 = 0 \quad \text{in } \Omega.
\end{aligned}$$

By classical arguments, there exists a constant $C > 0$ such that

$$\|\mathbf{Z}_0\|_{V(\Omega)} \leq C \left(\|\mathbf{G}\|_{V(\Omega)} + \|\mathbf{v}\|_{H^1(\Omega, \mathbb{R}^2)} \right),$$

and in view of the previous estimates and Poincaré inequality,

$$\|\mathbf{Z}_0\|_{V(\Omega)} \leq C \left(\|\mathbf{G}\|_{V(\Omega)} + \|s\|_{L^2(\Omega, \mathbb{R}^2)} \right).$$

Finally, define $\mathbf{Z} \in V(\Omega)$ by $\mathbf{Z} = \mathbf{Z}_0 + \mathbf{v}$. Then, the pair $(\mathbf{Z}, S)$ is the unique solution to the problem

$$D_{(\mathbf{U},R)} F(0, \mathbf{U}_0, R_0)(\mathbf{Z}, S) = (\mathbf{G}, s),$$

and satisfies the estimate

$$\|\mathbf{Z}\|_{V(\Omega)} + \|S\|_{L^2(\Omega)} \leq C \left(\|\mathbf{G}\|_{V(\Omega)} + \|s\|_{L^2(\Omega,\mathbb{R}^2)} \right).$$

By the implicit function theorem, there exists $T_0 > 0$ such that the mapping $t \in (-T_0, T_0) \mapsto (\mathbf{U}_t, R_t) \in V(\Omega) \times L^2(\Omega)$ is differentiable, and since the mapping $t \mapsto J_t$ is regular, we deduce that the mapping $t \in (-T_0, T_0) \mapsto (\mathbf{u}_{1,t} \circ \Phi_t, p_{1,t} \circ \Phi_t) \in V(\Omega) \times L^2(\Omega)$ is differentiable.

Second step: computation of the shape derivative. Assume that $\partial\Omega$ is of class $\mathcal{C}^2$. Notice that such assumption yield the existence of strong solutions of the involved partial differential equations. In particular, System (3.17) has a unique solution $(\mathbf{u}_\lambda, p_\lambda, \lambda)$ belonging to the space $[V(\Omega) \cap H^2(\Omega, \mathbb{R}^2)] \times H^1(\Omega) \times \mathbb{R}$.

Let $\mathbf{V} \in W^{3,\infty}(\mathbb{R}^2, \mathbb{R}^2)$. We introduce $T > 0$ and a mapping $t \in (-T, T) \mapsto \Phi_t \in W^{2,\infty}(\mathbb{R}^2, \mathbb{R}^2)$, of class $\mathcal{C}^3$, satisfying $\Phi_0 = \mathrm{Id}$ and $\frac{d\Phi_t}{d_t}_{|t=0} = \mathbf{V}$. Assume moreover that $\Phi_t(x) = x$ for every $x \in \partial\Omega \setminus \Gamma_1$ and every t. We set $\Omega_t := \Phi_t(\Omega)$ and define the shape derivative of J at Ω, in the direction $\mathbf{V}$, by

$$\langle dJ(\Omega), \mathbf{V} \rangle = \lim_{t \searrow 0} \frac{J(\Omega_t) - J(\Omega)}{t}.$$

Let us denote by $(\mathbf{u}'_\lambda, p'_\lambda)$ the Eulerian derivative of the pair $(\mathbf{u}_\lambda, p_\lambda)$, in other words the derivative of the mapping $t \mapsto (\mathbf{u}_{\lambda,t}, p_{\lambda,t}) \in H^1(\Omega) \times L^2(\Omega)$ at $t = 0$. It is rather standard to differentiate the partial differential equation (3.17) with respect to the domain perturbation. To that end, we need to introduce an extension of the normal vector. Recall that the final expression of the shape derivatives does not depend on the choice of extension (see [81, Theorem 5.9.2]).

Let us consider a symmetric extension $\mathbf{n}_t$ of the normal on $\partial\Omega_t$, in other words such that $\nabla \mathbf{n}_t$ is a symmetric matrix a.e. in $\overline{\Omega_t}$. Then the Eulerian derivative of this extension is given by

$$\frac{\partial \mathbf{n}_t}{\partial t} = \mathbf{n}' = -\nabla_\Gamma (\mathbf{V} \cdot \mathbf{n}) \tag{3.34}$$

where ∇_Γ is the tangential gradient operator and $\mathbf{n}'$ stands for the derivative of $t \mapsto \mathbf{n}_t$ at $t = 0$.

Using classical shape derivation rules [81, Sections 5.6 and 5.8], one gets

$$\begin{aligned}\langle dJ(\Omega), \mathbf{V}\rangle = \int_{\Gamma_1} \left[2\nu|e(\mathbf{u}_\lambda)|^2 + \beta\left(H|\mathbf{u}_\lambda|^2 + \partial_n(|\mathbf{u}_\lambda|^2)\right)\right](\mathbf{V}\cdot\mathbf{n})\,\mathrm{d}\mathcal{H}^1 \\ + \int_\Omega 4\nu e(\mathbf{u}_\lambda) : e(\mathbf{u}'_\lambda)\,dx + \int_{\Gamma_1} 2\beta\mathbf{u}_\lambda\cdot\mathbf{u}'_\lambda\,\mathrm{d}\mathcal{H}^1,\end{aligned} \tag{3.35}$$

where H denotes the mean curvature on $\partial\Omega$.

In order to get a more workable expression of this quantity (in view of numerical simulations), the general method is to introduce an adjoint problem to rewrite the term $4\nu\int_\Omega e(\mathbf{u}_\lambda) : e(\mathbf{u}'_\lambda)\,\mathrm{d}x + 2\beta\int_{\Gamma_1}\mathbf{u}_\lambda\cdot\mathbf{u}'_\lambda\,\mathrm{d}\mathcal{H}^1$ under the form $\int_\Gamma G(\mathbf{V}\cdot\mathbf{n})\,\mathrm{d}\mathcal{H}^1$, where G does not depend on $\mathbf{V}$. The shape derivative only depends and $\mathbf{u}_\lambda$ and p_λ, in particular it does not involve the solution of an adjoint state. The problem is thus "self-adjoint" in some sense. This result is expected since the criterion $J(\Omega)$ can be rewritten as the minimum of an energy functional.

In order to eliminate the last line from (3.35), let us use equation (3.17) and integrate by parts taking $\mathbf{u}'_\lambda$ as a test-function. We obtain

$$\begin{aligned}0 &= \int_\Omega -\operatorname{div}(\sigma(\mathbf{u}_\lambda, p_\lambda))\cdot\mathbf{u}'_\lambda\,dx \\ &= \int_\Omega 2\nu e(\mathbf{u}_\lambda) : e(\mathbf{u}'_\lambda)\,dx - \int_{\partial\Omega}\sigma(\mathbf{u}_\lambda,p_\lambda)\mathbf{n}\cdot\mathbf{u}'_\lambda\,\mathrm{d}\mathcal{H}^1 \\ &= \int_\Omega 2\nu e(\mathbf{u}_\lambda) : e(\mathbf{u}'_\lambda)\,dx - \int_{\Gamma_1}\sigma(\mathbf{u}_\lambda,p_\lambda)\mathbf{n}\cdot\mathbf{u}'_\lambda \\ &\quad - \int_{\Gamma_{in}}\sigma(\mathbf{u}_\lambda,p_\lambda)\mathbf{n}\cdot\mathbf{u}'_\lambda \\ &= \int_\Omega 2\nu e(\mathbf{u}_\lambda) : e(\mathbf{u}'_\lambda)\,dx + \int_{\Gamma_1}\beta\mathbf{u}_\lambda\cdot\mathbf{u}'_\lambda \\ &\quad - \int_{\Gamma_1}(\sigma(\mathbf{u}_\lambda,p_\lambda)\mathbf{n}\cdot\mathbf{n})(\mathbf{u}'_\lambda\cdot\mathbf{n}) - \lambda\int_{\Gamma_{in}}\mathbf{u}'_\lambda\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1.\end{aligned}$$

From the equality

$$\int_{\Gamma_{in}}\mathbf{u}_\lambda\cdot\mathbf{n}\,\mathrm{d}\mathcal{H}^1 = -\frac{Q}{2},$$

we infer (with a slight abuse of notation)

$$\left\langle d\int_{\Gamma_{in}} \mathbf{u}_\lambda \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1, \mathbf{V}\right\rangle = \int_{\Gamma_{in}} \mathbf{u}'_\lambda \cdot \mathbf{n}\,\mathrm{d}\mathcal{H}^1 = 0.$$

Finally, on Γ_1, one has

$$\mathbf{u}'_\lambda \cdot \mathbf{n} = -\partial_n(\mathbf{u}_\lambda \cdot \mathbf{n})(\mathbf{V}\cdot\mathbf{n}) + \mathbf{u}_\lambda \cdot \nabla_\tau(\mathbf{V}\cdot\mathbf{n}). \tag{3.36}$$

After rearranging the terms, one finally gets

$$\begin{aligned}\langle dJ(\Omega), \boldsymbol{\theta}\rangle = \int_{\Gamma_1} &\left[2\nu|e(\mathbf{u}_\lambda)|^2 + \beta(H|\mathbf{u}_\lambda|^2 + \partial_n(|\mathbf{u}_\lambda|^2))\right.\\ &\left.-2(\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}\cdot\mathbf{n})\partial_n(\mathbf{u}_\lambda\cdot\mathbf{n})\right](\mathbf{V}\cdot\mathbf{n})\,\mathrm{d}\mathcal{H}^1\\ &+\int_{\Gamma_1} 2(\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}\cdot\mathbf{n})\mathbf{u}_\lambda\cdot\nabla_\tau(\mathbf{V}\cdot\mathbf{n})\,\mathrm{d}\mathcal{H}^1;\end{aligned} \tag{3.37}$$

In what follows, we will denote by $\langle dJ(\Omega), \mathbf{V}\rangle$ the shape derivative of J at Ω in the direction $\mathbf{V}$, in other words

$$\langle dJ(\Omega), \mathbf{V}\rangle = \lim_{t\searrow 0} \frac{J(\Omega_t) - J(\Omega)}{t}.$$

From now on, we will assume at the same time more regularity on the domain Ω and on the vector field $\mathbf{V}$ in order to get a workable expression of the shape derivative. Hence, we will assume that $\partial\Omega$ is of class C^2 and $\mathbf{V} \in W^{3,\infty}(\mathbb{R}^2, \mathbb{R}^2)$. These properties ensure that the boundary of the domain Ω_t remains of class C^2, provided that t is small enough (see e.g. [54]).

Theorem 9. *Assume that $\partial\Omega$ is C^2. Let the triple $(\mathbf{u}_\lambda, p_\lambda, \lambda)$ be the unique solution to (3.8a)-(3.8b)-(3.8c). For every vector field $\mathbf{V} \in W^{3,\infty}$ having a compact support that does not intersect $\partial\Omega\backslash\Gamma_1$, there holds*

$$\langle dJ(\Omega), \mathbf{V}\rangle = \int_{\Gamma_1} j_1(\mathbf{V}\cdot\mathbf{n}) + j_2 \cdot \nabla_\Gamma(\mathbf{V}\cdot\mathbf{n})\,\mathrm{d}\mathcal{H}^1 \tag{3.38}$$

with

$$j_1 = 2\nu|e(\mathbf{u}_\lambda)|^2 + \beta\left(\partial_n(|\mathbf{u}_\lambda|^2) + H|\mathbf{u}_\lambda|^2\right) - 2(\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}\cdot\mathbf{n})\partial_n(\mathbf{u}_\lambda\cdot\mathbf{n}) \tag{3.39}$$

and

$$j_2 = 2(\sigma(\mathbf{u}_\lambda, p_\lambda)\mathbf{n}\cdot\mathbf{n})[\mathbf{u}_\lambda]_\tau \tag{3.40}$$

For the sake of clarity, the proof of Theorem 9 is postponed to section 3.3.2.

So far, we have detailed the sensitivity analysis with general domains. However, a shape-derivative expression is necessary when considering a symmetric domain Ω as in the numerical simulations (Section 3.4).

Corollary 10. *Let Ω as in Theorem 9, with the additional hypothesis that Ω is symmetric through hyperplane $\mathcal{H}$ and $(\mathbf{u}, p)$ belongs to $H^2(\Omega, \mathbb{R}^2) \times H^1(\Omega)$, the shape derivative of J_{sym} has the same expression as in Theorem 9, with a factor $\frac{1}{2}$, that is*

$$\langle dJ_{sym}(\Omega), \mathbf{V} \rangle = \frac{1}{2} \int_{\Gamma_1} j_1(\mathbf{V} \cdot \mathbf{n}) + j_2 \cdot \nabla_\Gamma(\mathbf{V} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 \tag{3.41}$$

Proof. Notice that using Property 6 (and in particular (3.20)), problem (3.24a)-(3.24b)-(3.24c) is problem (3.8a)-(3.8b)-(3.8c) where Γ_2 has been replaced by $\Gamma_2 \cup \Gamma_{sym}$. Thus proof of Corollary 10 is identical to that of Theorem 9 (appendix 3.3.2), replacing Γ_2 with $\Gamma_2 \cup \Gamma_{sym}$. □

3.4 Numerical methods and algorithms

In this section, we will take advantage of the tools developed in Section 3.3 to infer an efficient algorithm for solving Problem (3.27). The numerical developments proposed in the sequel rely on the `FreeFem++` [79] software, a free environment allowing to solve a wide variety of PDEs using the Finite Element method within a few command lines.

Let us first recall that in [73], the authors solved numerically a one-dimensional optimization problem, by assuming that each connected part of Γ_1 is a segment and making the inner angle between Γ_1 and Γ_2 vary. In a more recent article [19], shape optimization on hydrodynamic resistance is performed on a similar problem, assuming the cone wall is parametrized by a function depending on three parameters. A systematic (gradient-less) search is performed on the three parameters, and thus is computationally expensive.

In an attempt to improve the results mentioned above, we will enrich their approach by

- considering a wider class of admissible shapes for Γ_1,
- using numerical shape-optimization techniques based on the computation of the shape derivative.

We will solve the shape optimization problem (3.27), restricted the admissible shapes to symmetric ones, as stated in the last paragraph of Subsection 3.2.2. Notice that a close but simpler problem has been numerically investigated in [52]. In order to simplify notation, we present all the material on the full domain Ω, but the calculations remain valid on the symmetric problem (see Corollary 10), just by replacing Ω by Ω_+, Γ_1 by $\Gamma_1 \cap \mathcal{H}_+$, J by J_{sym}, etc.

Our approach can be decomposed into two main steps:

Step 1. Following [73], we recover the optimal inner angle between Γ_1 and Γ_2.

Step 2. Starting from the resulting straight cone with optimal angle, we find a local minimizer for the shape optimization problem (3.27), taking into account the symmetry constraint on the admissible shapes.

Let us stress that the proposed strategy is consistent with our main motivation, which is to reproduce the result presented in [73], and enhance the reduction of the cost functional by authorizing deformations of boundary Γ_1. Moreover, a global optimization of J with respect to Ω would require to handle the junction points between Γ_0 and Γ_1, where a transition occurs between no-slip and partial slip boundary conditions. To our best knowledge, due to this change of boundary conditions, the treatment of deformations of the boundary $\Gamma_0 \cap \Gamma_1$ would raise many difficulties, and would not be adapted to the tools developed within this article.

The two steps of the method are described in detail in the next two subsections.

3.4.1 Choice of parameters

In order to perform numerical tests, reasonable physical parameters ν, β, Q as well as geometric dimensions for the aquaporin are required. In what follows, we will use that when the initial geometry Ω is fixed, the shape-optimization problem depends only on the slip-length $b = \nu/\beta$.

To prove this, we introduce a normalized version of the energy, defined by

$$\tilde{\mathcal{E}}_\Omega(\mathbf{w}) = \frac{1}{\beta}\mathcal{E}_\Omega(\mathbf{w}) = \frac{\nu}{\beta}\int_\Omega |D(\mathbf{w})|^2\, dx + \frac{1}{2}\int_{\Gamma_1} |\mathbf{w}|^2\, d\mathcal{H}^2.$$

Ω being fixed, we see that the functional $\tilde{\mathcal{E}}_\Omega$ depends only the ratio ν/β. Recall that $\mathbf{u}_\lambda$ is defined as the minimizer of this energy over the space $V_{div}(\Omega)$:

$$\tilde{\mathcal{E}}_\Omega(\mathbf{u}_\lambda) = \min\{\tilde{\mathcal{E}}_\Omega(\mathbf{w}),\ \mathbf{w} \in V_{div}(\Omega)\}.$$

By this definition, it is clear that the minimum $\tilde{\mathcal{E}}_\Omega(\mathbf{u}_\lambda)$ depends only on ν/β. We claim that the same holds for the minimizer $\mathbf{u}_\lambda$. This is a consequence of the construction of $\mathbf{u}_\lambda$Indeed, using formulae (3.13)–(3.14) along with the definition of λ from Eq. (3.18), we can write

$$\lambda = -\frac{Q}{\int_{\Gamma_{in}} \mathbf{u}_1 \cdot \mathbf{n}\, d\mathcal{H}^1} = -\frac{2\beta\, \tilde{\mathcal{E}}_\Omega(\mathbf{u}_\lambda)}{Q}.$$

If Q is fixed, the previous expression shows that λ depends only on two parameters: β (and the dependence is linear), and the ratio ν/β. Finally, dividing Eq. (3.8a) by β cancels out the dependence on β, and we obtain a normalized equation whose coefficients depend only on the same ratio $b = \nu/\beta$. Thus, its solution $\mathbf{u}_\lambda$ depends only on this ratio.

Using the same kind of argument, we see that the pair $(\mathbf{u}_\lambda, p_\lambda)$ depends linearly on Q, so the shape-optimization problem (3.29) is independent of Q. Subsequently, we choose $Q = 1$.

Finally, following [74], we choose $L/a = 20$, where L is the length of the central tube, and a is the central tube radius.

In what follows, we will concentrate on two relevant test cases to present our numerical results. All the parameters are chosen to be relevant for practical issues (see Table 3.1).

	Test-case 1	Test-case 2
β	1	1
ν	20	100
τ_0	5×10^{-4}	2×10^{-4}
ε_{stop}	10^{-1}	10^{-1}
θ^*	0.265	0.311

Table 3.1 – *Model and numerical parameters. Note : $\theta^\star$ is a numerical result, obtained at step 1 of the optimization algorithm, see 3.4.2*

3.4.2 Finding the optimal angle (step 1 of the algorithm)

We reproduce here the analysis in [73]. The goal of the first step is to find the optimal angle between Γ_1 and Γ_2. In other words, we solve the following optimization problem

$$\boxed{\inf\{J(\Omega_\theta), \theta \in [0, \theta_{max})\}}, \tag{3.42}$$

Ω_θ being the domain Ω with angle θ between Γ_1 and Γ_2. The upper part of this domain, along with a computational mesh is depicted in Fig. 3.4 for $\theta \in \{0.1, 0.2, 0.4\}$.

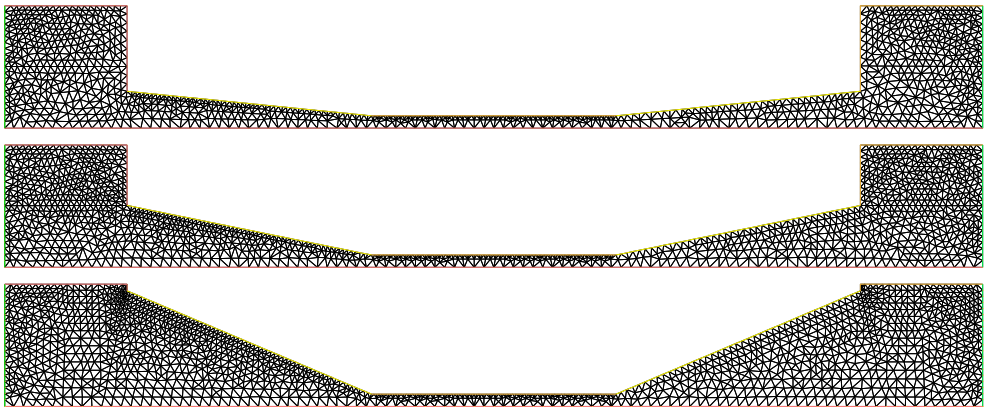


Figure 3.4 – *Computational domains for $\theta = 0.1, 0.2$ and 0.4, from top to bottom.*

For each value of θ, a simplicial mesh on Ω_θ is built, the Stokes equation (3.24a)–(3.24c) is solved using a standard finite elements

method (FEM). The velocity and pressure are respectively approximated by P_2 and P_1 elements. From this solution, an approximated value of $J(\Omega_\theta)$ is computed.

In order to solve numerically problem (3.42), a simple dichotomy like procedure is used, making the angle between Γ_1 and Γ_2 vary. Fig. 3.5 shows the graph of mapping $\theta \mapsto J(\Omega_\theta)$ for cases 1 and 2. In this case, using for example the golden section line search [32] gives $\theta^* = 0.265 \pm 0.001$, which is the value used in the second step. For test-case 2, $\theta^* = 0.311 \pm 0.001$.

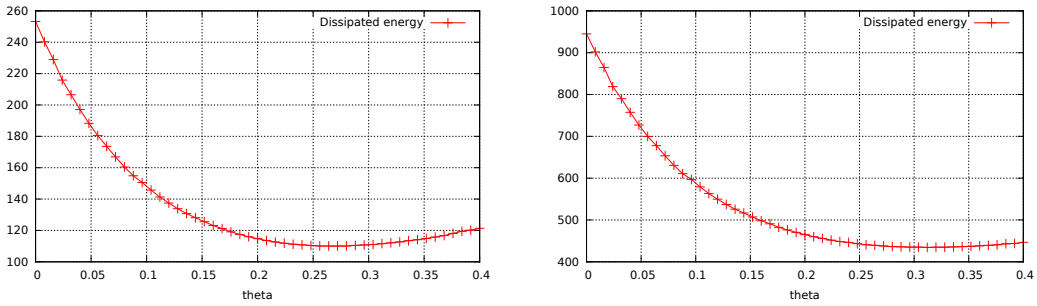


Figure 3.5 – *Criterion $J(\Omega)$ w.r.t. the angle parameter in Test-cases 1 and 2 (see Table 3.1).*

3.4.3 Optimizing the shape of Γ_1 (step 2 of the algorithm)

In what follows, we will consider polygonal shapes Ω, symmetric with respect to the hyperplane $\mathcal{H} = \{x_2 = 0\}$, and build a mesh $\mathcal{T}$ of their upper part $\Omega_+ = \Omega \cap \{x_2 > 0\}$, composed of K (closed) simplices $T_1, ..., T_K$ (i.e. triangles in $2d$, tetrahedra in $3d$), and I vertices $\mathbf{x}_1, ..., \mathbf{x}_I$. This mesh is assumed to be a conforming simplicial covering-up of Ω_+ [66].

Let us now provide the skeleton of the algorithm.

- **Initialization:** Choose an initial admissible domain Ω, symmetric with respect to $\mathcal{H}$. Define $\Omega^0 := \Omega_+$ as the upper part of Ω, and equip Ω^0 with a mesh $\mathcal{T}^0$.
- **For $n = 0$, ..., until convergence:**
  1. Compute the solution $(\mathbf{u}_\lambda^n, p_\lambda^n) \in \tilde{V}(\Omega^n) \times L^2(\Omega^n)$ of Stokes equation (3.24a)–(3.24c), in Ω^n, using the mesh $\mathcal{T}^n$.
  2. Compute the shape derivative of $J(\Omega^n)$ (see Corollary 10) and infer a descent direction $\boldsymbol{\theta}^n$ for the optimization problem (Section 3.4.3).
  3. Choose an appropriate gradient 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.

Convergence is reached whenever

$$\frac{|J(\Omega^{n+1}) - J(\Omega^n)|}{\tau^n} < \varepsilon_{stop}.$$

At each iteration, τ^n is initialized to a fixed value τ_0 and divided by $q = 1.5$ until

$$J((\mathrm{Id} + \tau^n \boldsymbol{\theta}^n)(\Omega^n)) < J(\Omega^n).$$

The mesh is finally advected point by point. Assuming the i^{th} point of Ω^n has coordinates $\mathbf{x}_i^n$,

$$\forall i \in \{1, \dots, I\}, \quad \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \tau^n \boldsymbol{\theta}^n(\mathbf{x}_i^n). \tag{3.43}$$

This procedure will result in a valid mesh only if $\boldsymbol{\theta}^n$ is smooth enough and τ is small enough. If this is not the case, self-intersections can appear. We address this difficulty with the extension procedure of our algorithm.

The choice of ε_{stop} has a minor impact on the final shape. Running the optimization algorithm on a fixed and large (typically 1000) number of steps, we noticed that

- Domains Ω^n converge to a fixed shape Ω^*.

- After many iterations, the value of J no longer decreases, no matter how small the gradient step τ_0. This is because the shape gradient approximately zero. Note that in this case, one does not necessarily have $j_1 = 0$ and $j_2 = 0$ but rather $j_1 - \mathrm{div}_\Gamma(j_2) = 0$ using Equation (3.48). Again, we do not use this last quantity as a stopping criterion since it is difficult to compute accurately due to the presence of high-order derivatives of $\mathbf{u}$.

Extension-regularization procedure

In this section, we present the core of step 2. The method presented here relies on a H^1-regularization step [57, 38, 51], followed by a linear elasticity-based extension. To the authors knowledge, it is not standard, and allows to take into account $\int_{\Gamma_1} j_2 \cdot \nabla_\tau(\boldsymbol{\theta} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1$ without assuming additional regularity on the term j_2.

Substep 1: regularization procedure. Let $U = H^1_0(\Gamma_1, \mathbb{R}^2)$ and let ϕ be the solution of the following PDE under variational form: find $\phi \in U$ such that for all $\psi \in U$,

$$\int_{\Gamma_1} \nabla_\Gamma \phi \cdot \nabla_\Gamma \psi \, \mathrm{d}\mathcal{H}^1 = -\int_{\Gamma_1} j_1 \psi + j_2 \cdot \nabla_\Gamma \psi \, \mathrm{d}\mathcal{H}^1 \qquad \text{for all } \psi \in U. \tag{3.44}$$

The existence of ϕ is standard, by Lax-Milgram theorem. Taking now $\psi = \phi$ as test function in (3.44) yields

$$\int_{\Gamma_1} j_1 \phi + j_2 \cdot \nabla_\Gamma \phi \, \mathrm{d}\mathcal{H}^1 = -\int_{\Gamma_1} |\nabla_\Gamma \phi|^2 \, \mathrm{d}\mathcal{H}^1 \leq 0. \tag{3.45}$$

Let us stress the importance of such a step, which provides a smooth function ϕ (in $H^1_0(\Gamma_1, \mathbb{R}^2)$) from the knowledge of j_1 and j_2 on Γ_1. Without this step, the algorithm produces increasingly distorted meshes, which are unsuitable for computation.

Substep 2: extension to the whole domain. We look for a vector field $\boldsymbol{\theta}$ satisfying at the same time

- $\boldsymbol{\theta} \cdot \mathbf{n} = \phi$ on Γ_1,
- and $\boldsymbol{\theta}$ is smooth inside Ω.

For that purpose, we choose $\boldsymbol{\theta} \in H^1(\Omega, \mathbb{R}^2)$ as the unique solution of the linear elasticity problem

$$\begin{cases} -\operatorname{div}(\sigma_e(\boldsymbol{\theta})) = 0 & \text{in} \quad \Omega \\ \boldsymbol{\theta} = 0 & \text{on} \quad \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_0 \\ \boldsymbol{\theta} \cdot \mathbf{n} = \phi & \text{on} \quad \Gamma_1 \\ [\sigma_e(\boldsymbol{\theta})\mathbf{n}]_\tau = 0 & \text{on} \quad \Gamma_1 \cup \Gamma_{sym} \\ \boldsymbol{\theta} \cdot \mathbf{n} = 0 & \text{on} \quad \Gamma_{sym} \end{cases} \tag{3.46}$$

where $\sigma_e(\boldsymbol{\theta})$ stands for the elasticity tensor given by

$$\sigma_e(\boldsymbol{\theta}) = \mu_e(\nabla\boldsymbol{\theta} + (\nabla\boldsymbol{\theta})^\top) + \lambda_e \operatorname{div}\boldsymbol{\theta}.$$

In practice, the parameters λ_e and μ_e are fixed respectively to 1 and 0.5.

This choice - which is widespread in meshing [18, 56, 50] 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).

It remains to show that, with the definitions above, the vector field $\boldsymbol{\theta}$ is a descent direction for J. According to Theorem 9, one has

$$\begin{aligned} \langle dJ(\Omega), \boldsymbol{\theta} \rangle &= \int_{\Gamma_1} j_1(\boldsymbol{\theta} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 + \int_{\Gamma_1} j_2 \cdot \nabla_\Gamma(\boldsymbol{\theta} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 \\ &= \int_{\Gamma_1} j_1\phi + j_2 \cdot \nabla_\Gamma\phi \, \mathrm{d}\mathcal{H}^1 = -\int_{\Gamma_1} |\nabla_\Gamma\phi|^2 \, \mathrm{d}\mathcal{H}^1, \end{aligned}$$

since the expression of the shape derivative only depends on $\boldsymbol{\theta} \cdot \mathbf{n}$ on Γ_1. We then infer that taking $\mathbf{V} = \boldsymbol{\theta}$ as the solution of system (3.46) provides a descent direction for J.

As pointed out in [57, Section 3.4], choosing a good inner product on Γ_1 is crucial for implementation and algorithmic efficiency issues. Note that, in that case, taking for instance a L^2 inner product may produce irregular domains. The choice of a H^1 inner product on the manifold Γ_1 offers a better alternative both for stability and convergence speed.

The extension step produces a displacement field defined on Ω^n. As mentioned in section 3.4.3, $\boldsymbol{\theta}^n$ needs to be smooth to avoid invalid

meshes. This is why the linear elasticity system is used (eq. (3.46)): the term $\lambda \operatorname{div}(\theta)$ is used to penalize local mesh compression, helping to avoid mesh self-intersections.

Let us conclude this paragraph with an important observation about the term $\int_{\Gamma_1} j_2 \cdot \nabla_\tau(\boldsymbol{\theta} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1$ appearing in the expression of the shape derivative of J. For smooth data $\boldsymbol{\theta}$, j_2 and Γ_1, the following integration by parts formula [81, Theorem 5.4.13] provides

$$\begin{aligned} \int_{\Gamma_1} j_2 \cdot \nabla_\Gamma(\theta \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 &= \int_\Gamma -(\theta \cdot \mathbf{n}) \operatorname{div}_\Gamma(j_2) + H(\theta \cdot \mathbf{n})(j_2 \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 \\ &= \int_\Gamma - \operatorname{div}_\Gamma(j_2)(\theta \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1. \end{aligned} \tag{3.47}$$

The last line is obtained by noticing that j_2 is contained in the tangent plane a.e. on Γ_1. Using Theorem 9, the expression of the shape derivative of J reduces to

$$\langle dJ, \boldsymbol{\theta} \rangle = \int_{\Gamma_1} (j_1 - \operatorname{div}_\Gamma(j_2))(\theta \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1. \tag{3.48}$$

This remark should normally allow to use traditional regularization methods, as described in [57]. However, as seen in Theorem 9, j_2 depends on first-order derivatives of $\mathbf{u}$ as well as the geometry of the domain. Formula (3.48) is therefore impractical for numerical purposes, since dealing with such a term would need to use high order finite elements and a very fine mesh, and would increase dramatically the cost of computation.

This is why the expression (3.48) is not directly used in the numerical algorithm we implemented. Notice that the term j_2 is in some sense regularized in the step (3.44).

3.4.4 Numerical results

From table 3.1, recall that $\nu = 20$ in test-case 1 whereas $\nu = 100$ in test-case 2. This means that the relative effect of volumic dissipation compared to surface shear friction is expected to be more important in test-case 2.

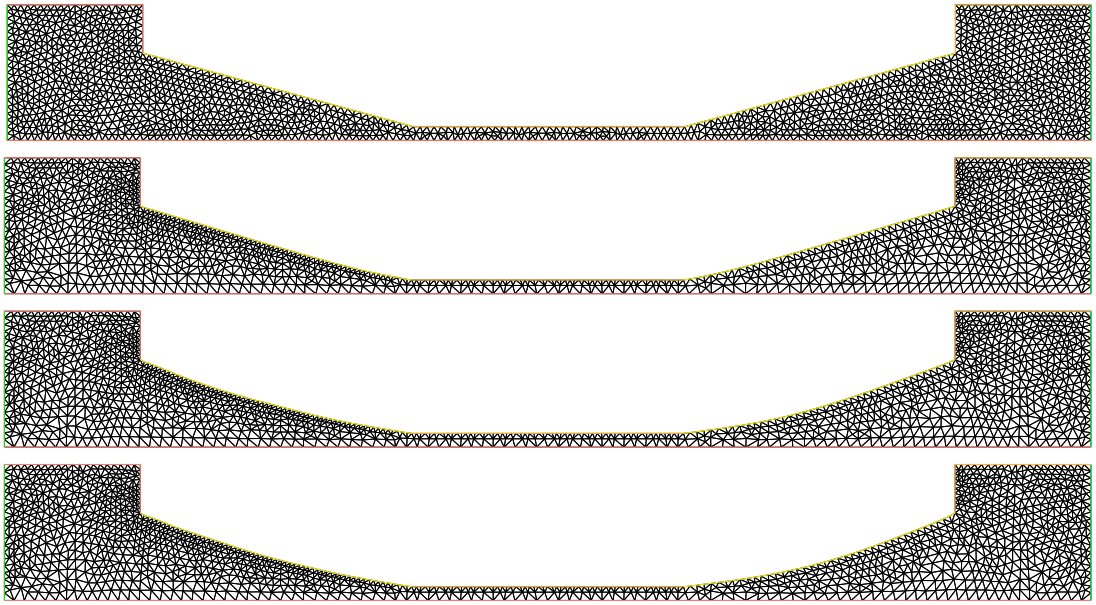


Figure 3.6 – *From top to bottom : mesh at iterations 0, 45 and 85 for Test-case 1.*

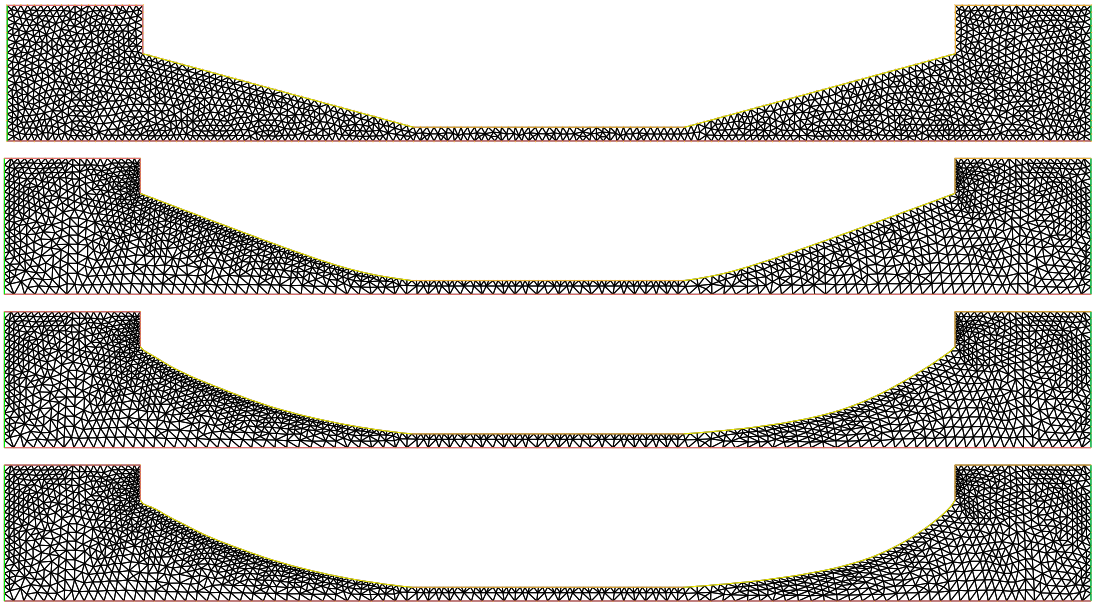


Figure 3.7 – *From top to bottom : mesh at iterations 0, 130 and 245 for Test-case 2.*

Comments. The step 2, described in Section 3.4.3, appears to be highly beneficial, leading to a 35% and 40% decrease of J for test-cases 1 and 2 respectively. Several aquaporin profiles along the

algorithm are displayed in Figures 3.6–3.7.

To take the analysis a step further, let us investigate which term in J contributes most to J between

- the viscous dissipation $J_d(\Omega) := 2\nu \int_\Omega |e(\mathbf{u}_{\Omega,\lambda})|^2 \, \mathrm{d}x$,

- the dissipation by friction $J_f(\Omega) := \beta \int_{\Gamma_1} |\mathbf{u}_{\Omega,\lambda}|^2 \, \mathrm{d}\mathcal{H}^1$.

Note that $J(\Omega) = J_d(\Omega) + J_f(\Omega)$.

In Figures 3.8–3.9, we observe that the reduction of the total dissipated energy achieved by step 2 of the algorithm, results from an important decrease of the viscous dissipation J_d, which appears to be the main contributor to the cost functional J. The frictional dissipation term J_f accounts for less than 10 percent of the total dissipation, and is slightly increased during the process.

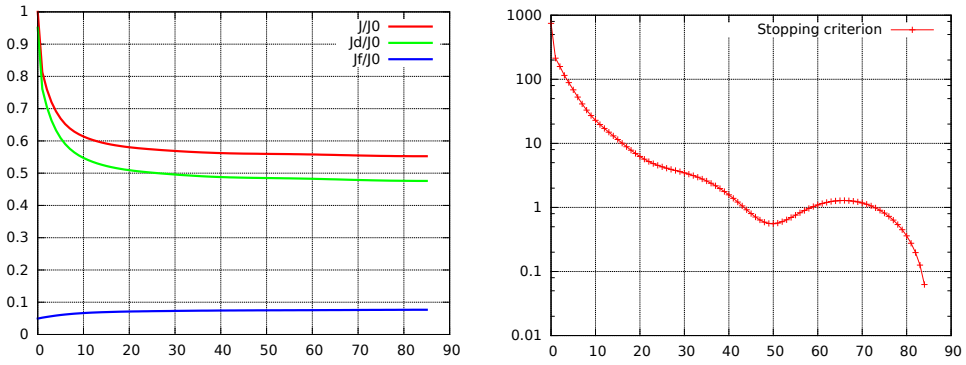


Figure 3.8 – *Test-case 1. Left: evolution of the cost functional J, and its sub-parts J_d, J_f over iterations, relatively to the initial value J_0 of J. Right: evolution of the stopping criterion.*

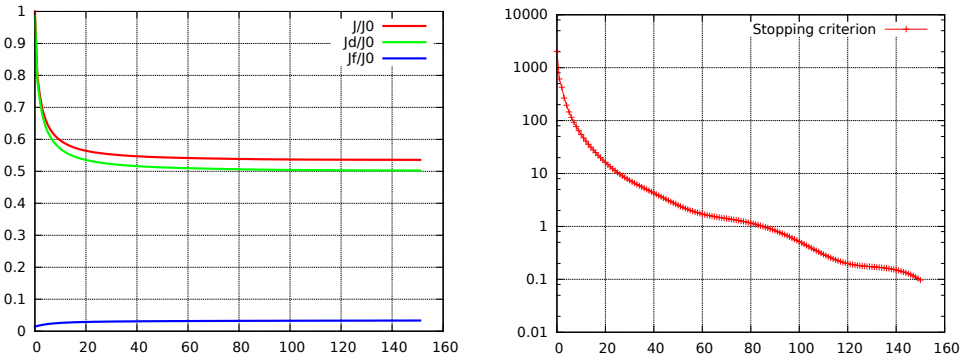


Figure 3.9 – *Test-case 2. Left: evolution of the cost functional J, and its sub-parts J_d, J_f over iterations, relatively to the initial value J_0 of J. Right: evolution of the stopping criterion.*

In what follows, we try to determine in which sub-domain of Ω the criterion J is decreased the most. For that purpose, one defines seven different regions as pictured on Fig. 3.10. On each of these regions R_i, we define

$$J_i(\Omega) = 2\nu \int_{R_i} |e(\mathbf{u}_{\Omega,\lambda})|^2 \, \mathrm{d}x + \beta \int_{\overline{R_i} \cap \Gamma_1} |\mathbf{u}_{\Omega,\lambda}|^2 \, \mathrm{d}\mathcal{H}^1, \quad i \in \{1, \dots, 7\}. \tag{3.49}$$

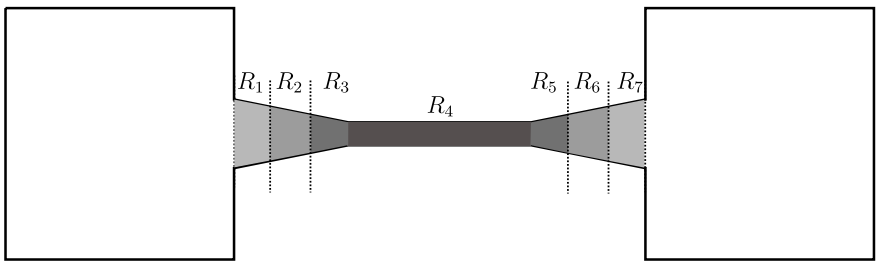


Figure 3.10 – *Outline of regions. Each region is defined as the intersection of Ω with an infinite vertical strip $\{a < x_1 < b\}$. Regions 1, 2 and 3 (resp. 5, 6, 7) each take one third of the inlet (resp. outlet) cone width, region 4 is the whole central tube.*

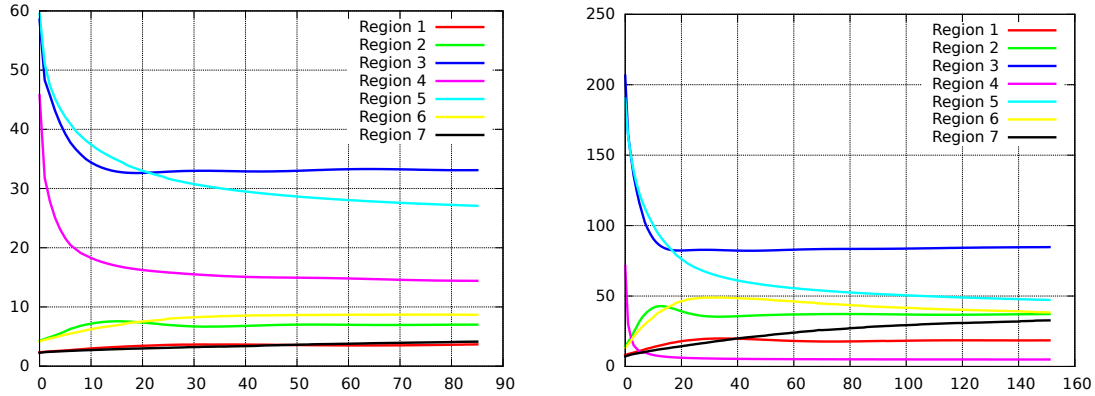


Figure 3.11 – *Evolution of each term J_i during step 2 of the optimization process. Test-case 1 on the left, 2 on the right.*

Note that J is not the sum of all terms J_i, the reservoirs being excluded. From the previous observations, considering only the J_d part of J provides a reasonable qualitative estimate of J, but the J_f part is also included for completeness. From Figure 3.11, it is visible that J is mostly decreased in the central tube and in regions close to it. This is expected, since this is where most dissipation takes place. On the contrary, J_i increases in other regions. This is not contradictory, since these regions only account for little dissipation. This can be seen as a trade-off to minimize the most important effects, resulting in a decreasing J in total. Finally, inlet and outlet dissipation (regions 1 and 7) looks almost unaffected. It may have been greatly decreased after step 1 (Section 3.4.2), leaving no space for further improvement.

3.5 Further comments and conclusion

On the initial value of θ and Step 1. As was highlighted in the description of the optimization process, the optimal shape is found by determining first the optimal angle θ^* between the aquaporin channel and the central part, and then by changing the profile of the channels (making the corresponding boundary vary without modifying its extremities). Figure 3.12 shows the importance of the first step. When choosing the best value for θ in

the first step (namely, θ^*), we observe that the criterion is smaller at initialization, but more importantly also at the end of the procedure.

Optimization and discretization of the problem In order to perform numerical experiments, it is necessary to choose a value for h, the typical mesh triangle size. If h is too large, the evaluation of J leads to numerical errors. On the other hand, the resolution of the Stokes equation becomes more expensive as h becomes low. In order to have a good balance, we start from a coarse mesh and refine the iterates during the algorithm. This strategy is widespread in the literature, and fully detailed in [109].

Conclusion We tried to enrich the approach presented in [74], in other words to improve the channel shapes that have been obtained and get a significant decrease of the objective function. Although we cannot compare precisely our results to those of [74] since an axisymmetric geometry was used therein and we used a planar geometry, Figure 3.5 reveals a similar (convex) profile for the function $\theta \mapsto J(\Omega_\theta)$, where Ω_θ is the domain with straight channels making an angle θ with the central part. As shown in Figures 3.8 and 3.9, our strategy led to decreasing the objective function by at least 40%, by comparison with the domain with straight channels and optimal angle θ^*.

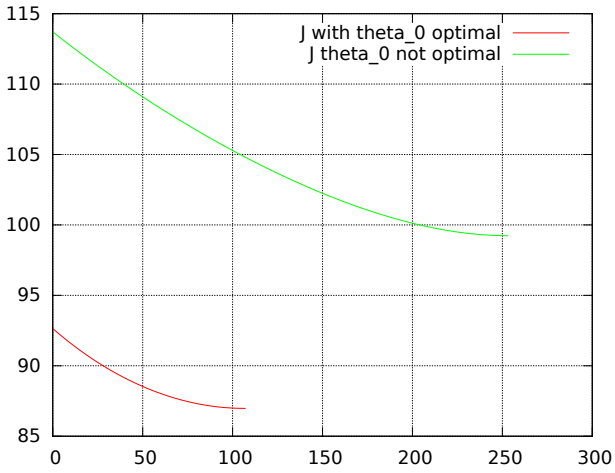


Figure 3.12 – *Plot of the objective functional during the iterations, starting with two different initial values for θ. The red curve is obtained with θ^* extracted from Table 3.1, the green curve is associated with $\theta = 0.3$. The final value of J is 21% lower when the initial angle θ^* is chosen.*

3.6 Shape-optimization of an axisymmetric aquaporin

3.6.1 Axisymmetric Stokes equations

The goal of this section is to prove that if Ω is smooth and invariant by rotation around an axis, then the weak solution of the Stokes problem 3.17 is also axisymmetric. First, we introduce some notations about axisymmetric domains and vector fields. **Notations** Let Ω an axisymmetric domain, *ie* a domain of $\mathbb{R}^3$ such that for all $\eta \in \mathbb{R}$,

$$R_\eta(\Omega) = \Omega$$

where R_η is the rotation of angle η around axis z. Let Γ_{in} and Γ_{out} two axisymmetric domains included in hyperplanes $z = 0$ and $z = L$ respectively. We make the following assumptions

- $\Gamma_{\text{in}} \subset \partial\Omega$ and $\Gamma_{\text{out}} \subset \partial\Omega$, and $\Gamma := \partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}})$ does not intersect hyperplanes $z = 0$ and $z = L$.
- Ω is included in $\{(x, y, z) \in \mathbb{R}^3, 0 \leq z \leq L\}$

- Γ is Lipschitz

Let $\mathbf{u} : \bar{\Omega} \to \mathbb{R}^3$ a smooth function.

We define the rotated version of $\mathbf{u}$ to angle η as $\mathbf{v} = R_{-\eta} \circ \mathbf{u} \circ R_\eta$. Note the double composition, the first one to transport the vector, the second one to rotate it.

Where no confusion is possible, we denote for any vector field f

$$R_\eta \circ f = R_\eta f$$

Definition 4. *Let* $\mathbf{u}$ *a vector field defined on* Ω*. Denoting* $\mathbf{u} = u_r \mathbf{e}_r + u_\theta \mathbf{e}_\theta + u_z \mathbf{e}_z$*,* $\mathbf{u}$ *is axisymmetric if it satisfies the two equations :*

$$\begin{aligned} R_{-\eta}\mathbf{u} \circ R_\eta = \mathbf{u} \quad \forall \eta \in \mathbb{R} \\ \mathbf{u}_\theta = 0 \end{aligned} \tag{3.50}$$

3.6.2 Energy formulation of the Stokes problem

Let $Q \neq 0$. We consider the energy minimization problem from Section 3.2. Let us recall energy minimization problem defined in Equation (3.11).

$$\begin{aligned} \min_{\mathbf{u} \in H^1(\Omega)^d} \quad & \mathcal{E}(\mathbf{u}) \\ \mathbf{u} \cdot n = 0 \text{ on } \Gamma \\ \textstyle\int_{\Gamma_{\text{in}}} \mathbf{u} \cdot n \, \mathrm{d}s = Q \\ \operatorname{div}(\mathbf{u}) = 0 \text{ on } \Omega \end{aligned} \tag{3.51}$$

where

$$\mathcal{E}(\mathbf{w}) = \nu \int_\Omega |D(\mathbf{w})|^2 \, \mathrm{d}x + \frac{\beta}{2} \int_\Gamma |\mathbf{w}|^2 \, \mathrm{d}s \tag{3.52}$$

3.6.3 Result of symmetry

In Proposition 5, we proved that Problem (3.51) has a unique solution, and characterized it with a weak formulation. Using this uniqueness property, we now prove that if Ω is axisymmetric, then so is the solution of the Stokes problem on Ω.

Theorem 11. *Let* $\mathbf{u}$ *solution to* (3.51)*. Then* $\mathbf{u}$ *is axisymmetric.*

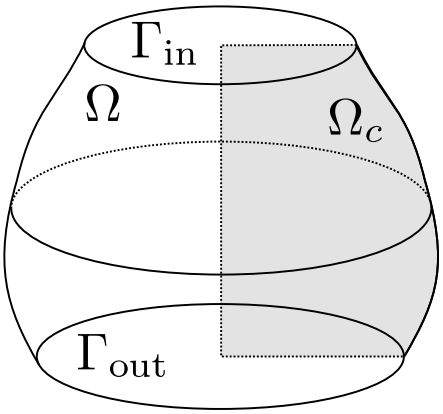


Figure 3.13 – Illustration of the meridian located at $\theta = 0$.

Remark 11. *The first equality of Theorem 11 also writes*

$$\begin{aligned} u_r(r,\eta,z) &= u_r(r,0,z) \\ u_\theta(r,\eta,z) &= u_\theta(r,0,z) \quad \forall \eta \in \mathbb{R} \\ u_z(r,\eta,z) &= u_z(r,0,z) \end{aligned}$$

For the proof of Theorem 11, the reader is referred to Appendix C.

3.6.4 Variational formulation of the problem in a meridian

It has been established in Theorem 11 that the flow is axisymmetric. This result allows us to reduce the original 3D problem to a 2D problem, thus also reducing the computational cost of numerical experiments. In this section, we derive a mixed velocity-pressure variational formulation for the Stokes equations from the energy expression $\mathcal{E}(\mathbf{u})$. For an axisymmetric flow, $u_\theta = 0$, therefore $\mathcal{E}(\mathbf{u}) = \mathcal{E}_1(\mathbf{u})$. Furthermore, we will restrict $\mathbf{u}$ to its value on the $\theta = 0$ meridian (see Fig. 3.13):

$$\Omega_c = \{(r,0,z),(r,\theta,z) \in \Omega\} \tag{3.53}$$

To take into account the $\frac{1}{r}$ singularity and r factor in $\mathcal{E}_1$, we introduce weighted Sobolev spaces. The variational formulation of

the Stokes equations is detailed in e.g [25]. We recall it here for the sake of completeness.

$$L^2_{\pm 1}(\Omega_c) = \left\{ v : \Omega_c \to \mathbb{R} \text{ measurable }, \int_{\Omega_c} |v(r,z)|^2 r^{\pm 1} \, \mathrm{d}r \, \mathrm{d}z < +\infty \right\}$$

,

$$H^1_1(\Omega_c) = \left\{ v \in L^2_{-1}(\Omega_c), \text{ such that } \frac{\partial v}{\partial r} \in L^2_1(\Omega_c), \frac{\partial v}{\partial z} \in L^2_1(\Omega_c) \right\}$$

The radial component (resp. axial component, resp. pressure) function space is defined as

$$\begin{array}{rcl} V_r(\Omega_c) & = & L^2_{-1}(\Omega_c) \cap H^1_1(\Omega_c) \\ V_z(\Omega_c) & = & H^1_1(\Omega_c) \\ M(\Omega_c) & = & L^2_1(\Omega_c) \end{array}$$

The velocity space is

$$V = \{ \mathbf{u} = (u_r, u_z) \in V_r \times V_z \text{ such that } \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma_c \}$$

The variational formulation of the problem is given by Find $(\mathbf{u}, p, \lambda) \in V \times M \times \mathbb{R}$ such that for all $(\mathbf{v}, q) \in V \times M$,

$$\left\{ \begin{array}{lcl} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) & = & 0 \\ b(\mathbf{u}, q) & = & 0 \\ F(\mathbf{u}) & = & Q \end{array} \right. \tag{3.54}$$

where $a : V \times V \to \mathbb{R}$ is defined as

$$\begin{aligned} a((u_r, u_z), (v_r, v_z)) = 2\nu & \int_{\Omega_c} \frac{\partial u_r}{\partial r} \frac{\partial v_r}{\partial r} + \frac{u_r}{r} \frac{v_r}{r} + \frac{\partial u_z}{\partial z} \frac{\partial v_z}{\partial z} \, r \, \mathrm{d}r \, \mathrm{d}z \\ & + \nu \int_{\Omega_c} \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r} \right) r \, \mathrm{d}r \, \mathrm{d}z \\ & + \beta \int_{\Gamma_c} [\mathbf{u}]_\tau \cdot [\mathbf{v}]_\tau \, r \, \mathrm{d}s \end{aligned} \tag{3.55}$$

and $b : V \times M \to \mathbb{R}$ is defined as

$$b((u_r, u_z), p) = \int_{\Omega_c} p \left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} \right) r \, \mathrm{d}z \tag{3.56}$$

Finally, the flux through Γ_{in} is defined as

$$F(\mathbf{u}) = \int_{\Gamma_{\text{in}}} r u_z \, \mathrm{d}r \tag{3.57}$$

In[25, Chapter IX], the well-posedness of a problem very similar to (3.54) (with Dirichlet boundary conditions on Γ_c) is proven using the LBB condition for saddle-points. The same arguments apply here, we do not repeat the proof.

3.6.5 Shape derivative of the energy functional

In section 3.3, we gave an expression for the derivative of the energy. We claim that this shape derivative can be translated to a meridian for an axisymmetric. Indeed, to each $\mathbf{u}_c : \Omega_c \to \mathbb{R}^2$ corresponds a unique axisymmetric extension to Ω $\mathbf{u} : \Omega \to \mathbb{R}^3$ defined as

$$E_\Omega(\mathbf{u}_r) = \mathbf{u}(r, \theta, z) = R_\theta \mathbf{u}_c(r, z)$$

where R_θ is the rotation matrix of angle θ around axis z. From the definition of $\mathcal{E}_1$ and J,

$$2\pi \mathcal{E}_1(\Omega_c) = J(\Omega)$$

Theorem 12. *The expression for the shape derivative of $\mathcal{E}_1$ is given by*

$$\langle d\mathcal{E}_1(\Omega_c), \mathbf{V} \rangle = \frac{1}{2\pi} \int_{\Gamma_1} j_1(\mathbf{V} \cdot \mathbf{n}) + j_2 \cdot \nabla_\Gamma(\mathbf{V} \cdot \mathbf{n}) \, \mathrm{d}\mathcal{H}^1 \tag{3.58}$$

with

$$\begin{aligned} j_1 &= 2\nu |e(\mathbf{u})|^2 + \beta \left(\partial_n(|\mathbf{u}|^2) + H_c |\mathbf{u}|^2 \right) - 2(\sigma(\mathbf{u}, p_\lambda)\mathbf{n} \cdot \mathbf{n}) \partial_n(\mathbf{u} \cdot \mathbf{n}) \\ j_2 &= 2(\sigma(\mathbf{u}, p_\lambda)\mathbf{n} \cdot \mathbf{n})[\mathbf{u}]_\tau \end{aligned}$$

The 3D mean curvature H_c on Γ_c is given by

$$H_c = \frac{1}{2} \left(H_{2D} + \frac{n_r}{r} \right)$$

where H_{2D} is the mean curvature of Γ.

Proof. Theorem 12 is an application of Theorem 9 in the case where Ω is axisymmetric. Let us focus on the expression of the mean curvature. Details of the following proof are given in [2, Chapter 15]. Supposing Γ_c is parametrized by

$$\begin{cases} x(t,\phi) &= r(t)\cos(\phi) \\ y(t,\phi) &= r(t)\sin(\phi) \\ z(t,\phi) &= h(t) \end{cases} \tag{3.59}$$

with $t \in [0,T]$ and $\phi \in (-\pi,\pi]$, the principal curvatures at (t,ϕ) are given by

$$\begin{cases} \kappa_1 &= \frac{-r''(t)h'(t)+r'(t)h''(t)}{(r'(t)^2+h'(t)^2)^{\frac{3}{2}}} \\ \kappa_2 &= \frac{h'(t)}{r(t)(r'(t)^2+h'(t)^2)^{\frac{1}{2}}} \end{cases}. \tag{3.60}$$

On the other hand, the normal vector is given by

$$\mathbf{n}(t,\phi) = \frac{1}{(r'(t)^2+h'(t)^2)^{\frac{1}{2}}}\left(h'(t)\mathbf{e}_r - r'(t)\mathbf{e}_z\right) \tag{3.61}$$

Therefore, $\kappa_2 = \frac{n_r}{r}$. In κ_1 from Equation (3.60), one recognizes the curvature of the planar curve Γ parametrized by ($t \in [0,T]$)

$$\begin{cases} x(t,\phi) &= r(t) \\ y(t,\phi) &= 0 \\ z(t,\phi) &= h(t) \end{cases} \tag{3.62}$$

which we called H_{2D} in the statement of the theorem. In other words, $\kappa_1 = H_{2D}$. Finally, $H_c = \frac{\kappa_1+\kappa_2}{2}$, which gives the desired result. □

3.6.6 First numerical results

In this subsection, we present some preliminary numerical results. The mixed formulation (3.54) is discretized with stable P2 (velocity)-P1 (pressure) finite elements on an unstructured mesh. Figure 3.14 features an example of axisymmetric velocity and pressure profile. We expect that Theorem 12, combined with an extension-regularization procedure (see Section 3.4.3) allows to find a descent

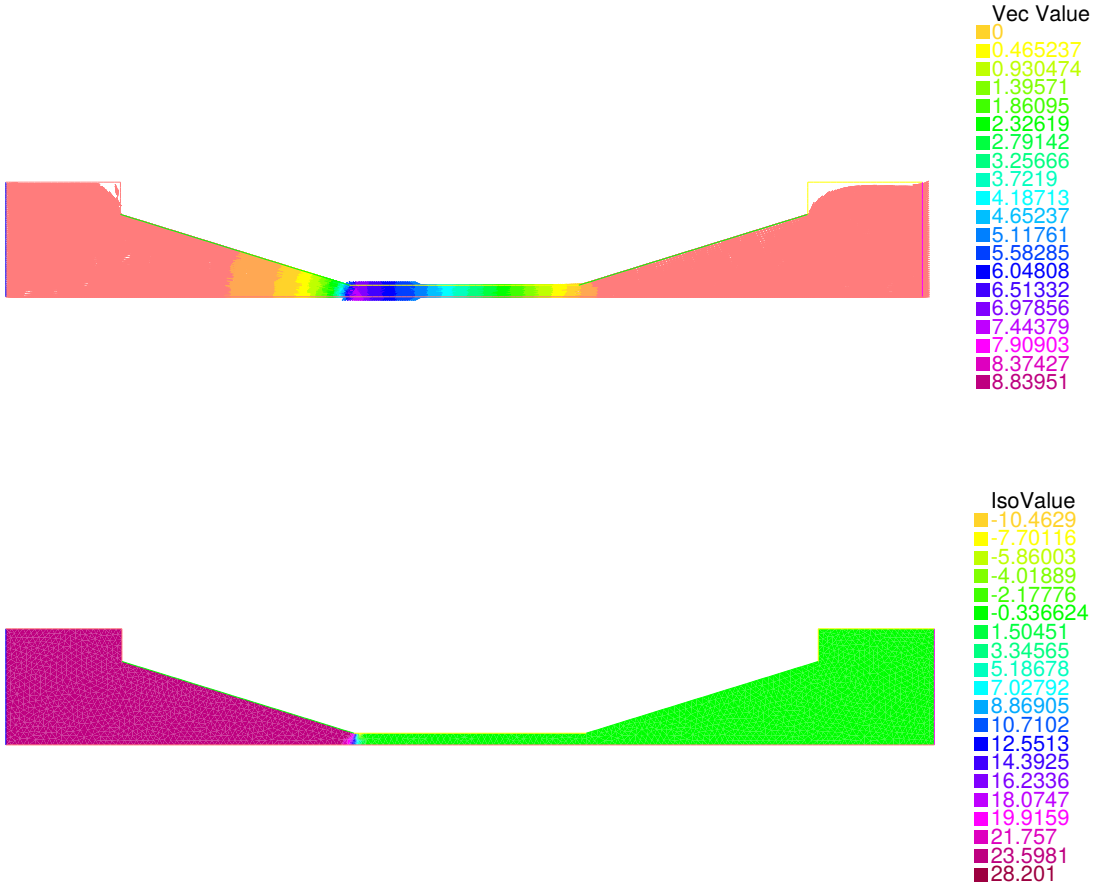


Figure 3.14 – *Velocity and pressure profiles on an axisymmetric domain*

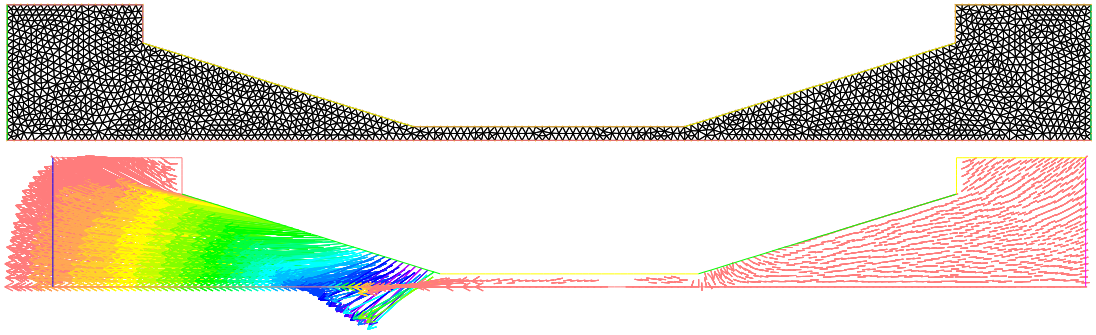


Figure 3.15 – *Top : initial mesh. Bottom : displacement field associated to the first iteration*

$J(\Omega_0)$	26.019
$J(\Omega_1)$	26.0123

Table 3.2 – *Functional value for the initial domain and after the first iteration.*

direction to be implemented within a gradient descent method as earlier. Figure 3.15 displays the initial domain and the displacement field corresponding to the first iteration. The first iteration shows a decrease of the criterion (Table 3.2), but unfortunately not in the next iterations.

This is a preliminary work in collaboration with François Detcheverry (ILM Lyon 1). The goal is to obtain more physically realistic results : in Section 3.4, only planar aquaporins were considered. It is more accurate from a physical point of view to consider an axisymmetric aquaporin.

Chapter 4

Conclusion and perspectives

In this thesis manuscript, we intended to study a variety of numerical geometry optimization problems in relation with incompressible fluids.

Chapter 1

The first chapter of this work was inspired by the two following papers :

- [13] using FreeFem++ to perform geometry optimization for elasticity problems
- [130] developing a method and offering a simple code for topology optimization for elasticity problems

These works have inspired us to present a relatively simple framework dealing with geometry optimization problems associated with fluids. We have designed and run several numerical examples, suggesting that the algorithm and its implementation can effectively solve shape optimization problems for several functionals and geometrical settings. The information provided in this chapter makes it possible to adapt the case to other equations, functionals and geometries.

In running `optiflow`, we have noticed some limitations. We list them below as well as potential solutions or workarounds.

- Change of topology. In Section 2.5.6, if the wrong parameters are chosen, the obstacle in the fluid may collapse. The resulting mesh is invalid : it does not represent a physical geometry, and it is not suitable for FEM computations. The workaround we use is to carefully choose the parameters ℓ and b^0 to avoid any collapse. We know of no solution if the problem intrinsically involves changes in topology, other than initializing with the right topology.

- Slow convergence. Because we use a first-order optimization method (based on the gradient method), convergence can be considered slow. One solution could be to use second-order methods involving the Hessian of the functional, see [4].

Finally, all the theory presented in Chapter 1 can be applied to three-dimensional settings. Apart from the resolution cost of the Navier-Stokes equations, the main numerical difficulty we expect is mesh advection. In particular, topology changes may be more natural in a 3-D setting, and should be carefully handled.

Chapter 2

The starting point of Chapter 2 is papers [74, 73]. Our idea was to use the same model and functional but to extend the class of admissible domains by using more design parameters.

First, we studied the proposed modelling of an aquaporin involving partial sleep boundary conditions. We made fluid model precise and proved its well-posedness, then proved that the shape optimization problem has a solution in a class of uniformly smooth domains.

As we emphatized in Theorem 9, the shape derivative has a different structure than that in Chapter 1 with no-slip boundary conditions. We introduced a specific numerical treatment for this difficulty.

Numerical results indicate that our method provides promising

results : starting from the optimal angle, we measured that the energy criterion is decreased by over 40% at the end of algorithm.

We started a joint work with F. Detcheverry (ILM, Lyon 1) on shape optimization assuming the sought domain is axisymmetric.

Appendices

A Computation of shape derivatives

This section is devoted to proving Theorem 2, in other words to the computation of the shape derivatives of the functionals E and D.

Green formula are required to perform integrations by part. For the sake of clarity, we recall them here.

Lemma 13. *Let $A \in H^1(\Omega)^{d\times d}$ and $\mathbf{v} \in H^1(\Omega)^d$ a matrix and a vector field defined over a Lipschitz set Ω. Then*

$$\int_\Omega A : \nabla(\mathbf{v})\, dx = \int_{\partial\Omega} A\mathbf{n}\cdot\mathbf{v}\, ds - \int_\Omega \operatorname{div}(A)\cdot\mathbf{v}\, dx$$

Where $\mathbf{n}$ is the outward-pointing normal vector.

Let $\mathbf{y}$ and $\mathbf{z}$ belong to $H^1(\Omega)^d$. Then

$$2\int_\Omega e(\mathbf{z}) : e(\mathbf{y})\, dx = -\int_\Omega (\triangle\mathbf{z} + \nabla \operatorname{div}\mathbf{z})\cdot\mathbf{y}\, dx + 2\int_{\partial\Omega} e(\mathbf{z})\mathbf{n}\cdot\mathbf{y}\, ds.$$

These relations are easily proven by decomposing the left-hand side, applying $d\times d$ times the scalar Green formula and re-combining all terms to obtain the right-hand side.

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 [81] for the rigorous definition of this notion, and to [83] or [53] for this precise calculation. A proof of differentiability and formulae for general shape functionals are provided in [127]. The derivative $(\mathbf{u}', p')$ of $(\mathbf{u}, p)$ with respect to the domain, in the direction $\boldsymbol{\theta} \in \Theta_{ad}$,

is solution to the problem:

$$\left\{\begin{array}{rcll} -\nu\Delta\mathbf{u}' + (\nabla\mathbf{u})\mathbf{u}' + (\nabla\mathbf{u}')\mathbf{u} + \nabla p' &=& 0 & x \in \Omega, \\ \operatorname{div}(\mathbf{u}') &=& 0 & x \in \Omega, \\ \sigma(\mathbf{u}', p')\mathbf{n} &=& 0 & x \in \Gamma_{\text{out}}, \\ \mathbf{u}' &=& 0 & x \in \Gamma_{\text{in}}, \\ \mathbf{u}' &=& -\left(\frac{\partial \mathbf{u}}{\partial n}\right)(\boldsymbol{\theta}.n) & x \in \Gamma. \end{array}\right. \tag{1}$$

A.1 Shape derivative of the functional E

Using Green's formula in (1), one gets

$$\begin{aligned} E'(\Omega)(\boldsymbol{\theta}) &= 4\nu \int_\Omega e(\mathbf{u}) : e(\mathbf{u}')\, dx + 2\nu \int_\Gamma |e(\mathbf{u})|^2\, \boldsymbol{\theta}\cdot\mathbf{n}\, ds \\ &= -2\nu \int_\Omega ((\triangle\mathbf{u} + \nabla \operatorname{div}\mathbf{u})\cdot\mathbf{u}')\, dx + 4\nu \int_{\partial\Omega} e(\mathbf{u})\mathbf{n}\cdot\mathbf{u}'\, ds \\ &\quad +2\nu \int_{\partial\Omega} |e(\mathbf{u})|^2\, \boldsymbol{\theta}\cdot\mathbf{n}\, ds \end{aligned}$$

Now, let us multiply the first equation of the adjoint problem (2.11) by $\mathbf{u}'$ and integrate over Ω, one obtains

$$\begin{aligned} -\nu \int_\Omega \triangle\mathbf{v}_e\cdot\mathbf{u}'\, dx + \int_\Omega \nabla q\cdot\mathbf{u}'\, dx + \int_\Omega (\nabla\mathbf{u})^T\mathbf{v}_e\cdot\mathbf{u}'\, dx \\ - \int_\Omega (\nabla\mathbf{v}_e)\mathbf{u}\cdot\mathbf{u}'\, dx = -2\nu \int_\Omega \triangle\mathbf{u}\cdot\mathbf{u}'\, dx. \end{aligned}$$

Using one integration by parts and the boundary conditions satisfied by $\mathbf{u}'$ and $\mathbf{v}_e$, we get

$$\begin{aligned} \int_\Omega \left(2\nu e(\mathbf{u}') : e(\mathbf{v}_e) - (\nabla\mathbf{v}_e)\mathbf{u}'\cdot\mathbf{u} + (\nabla\mathbf{u}')\mathbf{u}\cdot\mathbf{v}_e\right) dx \\ - \int_{\Gamma_{\text{out}}} \sigma(\mathbf{v}_e, q)\mathbf{n}\cdot\mathbf{u}'\, ds + \int_{\Gamma_{\text{out}}} \left((\mathbf{u}\cdot\mathbf{v}_e)(\mathbf{u}'\cdot\mathbf{n}) - (\mathbf{u}\cdot\mathbf{n})(\mathbf{u}'\cdot\mathbf{v}_e)\right) ds \\ - \int_\Gamma \sigma(\mathbf{v}_e, q)\mathbf{n}\cdot\mathbf{u}'\, ds = -2\nu \int_\Omega \triangle\mathbf{u}\cdot\mathbf{u}'\, dx. \end{aligned}$$

In the same way, if we multiply the first equation of the problem (1) by $\mathbf{v}_e$ and integrate over Ω, we obtain

$$-\nu \int_\Omega \triangle\mathbf{u}'\cdot\mathbf{v}_e\, dx + \int_\Omega \nabla p'\cdot\mathbf{v}_e\, dx + \int_\Omega (\nabla\mathbf{u}')\mathbf{u}\cdot\mathbf{v}_e\, dx + \int_\Omega (\nabla\mathbf{u})\mathbf{u}'\cdot\mathbf{v}_e\, dx = 0$$

and

$$\int_\Omega (2\nu e(\mathbf{u}') \cdot e(\mathbf{v}_e) + (\nabla \mathbf{u}')\mathbf{u} \cdot \mathbf{v}_e - (\nabla \mathbf{v}_e)\mathbf{u}' \cdot \mathbf{u})\, dx + \int_{\Gamma_{\text{out}}} (-\sigma(\mathbf{u}', p')\mathbf{n} \cdot \mathbf{v}_e + (\mathbf{u} \cdot \mathbf{v}_e)(\mathbf{u}' \cdot \mathbf{n}))\, ds = 0.$$

Coming back to the shape derivative expression

$$\begin{aligned} E'(\Omega)(\theta) &= -2\nu \int_\Omega ((\triangle \mathbf{u} + \nabla \operatorname{div} \mathbf{u}) \cdot \mathbf{u}')\, dx + 4\nu \int_{\partial\Omega} e(\mathbf{u})\mathbf{n} \cdot \mathbf{u}'\, ds \\ &\quad + 2\nu \int_{\partial\Omega} |e(\mathbf{u})|^2 (\boldsymbol{\theta} \cdot \mathbf{n})\, ds \\ &= A + 4\nu \int_{\partial\Omega} e(\mathbf{u})\mathbf{n} \cdot \mathbf{u}'\, ds + 2\nu \int_{\partial\Omega} |\varepsilon(\mathbf{u})|^2 (\boldsymbol{\theta} \cdot \mathbf{n})\, ds, \end{aligned}$$

where we set $A := -2\nu \int_\Omega ((\triangle \mathbf{u} + \nabla \operatorname{div} \mathbf{u}) \cdot \mathbf{u}')\, dx$. Using the previous identities, we get for A

$$A = \int_{\Gamma \cup \Gamma_{\text{out}}} (q\mathbf{n} - 2\nu e(\mathbf{v}_e)\mathbf{n}) \cdot \mathbf{u}'\, ds - \int_{\Gamma_{\text{out}}} (\mathbf{u} \cdot \mathbf{n})(\mathbf{v}_e \cdot \mathbf{u}')\, ds.$$

Therefore, according to (2.11)

$$\begin{aligned} E'(\Omega)(\theta) &= \int_{\Gamma \cup \Gamma_{\text{out}}} (q\mathbf{n} - 2\nu e(\mathbf{v}_e)\mathbf{n}) \cdot \mathbf{u}'\, ds - \int_{\Gamma_{\text{out}}} (\mathbf{u} \cdot \mathbf{n})(\mathbf{v}_e \cdot \mathbf{u}')\, ds \\ &\quad + 4\nu \int_{\Gamma_{\text{out}} \cup \Gamma} e(\mathbf{u})\mathbf{n} \cdot \mathbf{u}'\, ds + 2\nu \int_\Gamma |e(\mathbf{u})|^2\, \boldsymbol{\theta} \cdot \mathbf{n}\, ds \\ &= \int_\Gamma (q\mathbf{n} - 2\nu e(\mathbf{v}_e)\mathbf{n} + 4\nu e(\mathbf{u})\mathbf{n}) \cdot \mathbf{u}'\, ds + 2\nu \int_\Gamma |e(\mathbf{u})|^2\, \boldsymbol{\theta} \cdot \mathbf{n}\, ds \\ &= -\int_\Gamma \left((q\mathbf{n} - 2\nu e(\mathbf{v}_e)\mathbf{n} + 4\nu e(\mathbf{u})\mathbf{n}) \cdot \frac{\partial \mathbf{u}}{\partial n} - 2\nu |e(\mathbf{u})|^2 \right) \boldsymbol{\theta} \cdot \mathbf{n}\, ds \end{aligned}$$

To get a more symmetric expression, one can use the following elementary properties: since $\mathbf{u}$ and $\mathbf{v}_e$ are divergence-free and vanish on Γ, we have on this boundary

- $\mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial n} = 0$.
- $e(\mathbf{u}) \cdot \mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = |e(\mathbf{u})|^2$.
- $(e(\mathbf{v}_e) \cdot \mathbf{n}) \cdot \frac{\partial \mathbf{u}}{\partial n} = e(\mathbf{u}) : e(\mathbf{v}_e)$.

The expected result follows.

A.2 Shape derivative of the functional D

Using the chain rule from the definition (2.4) of $D(\Omega)$ yields:

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

The main idea of the proof consists in using the adjoint state $(\mathbf{v}_d, q_d)$, solution to (2.13): performing several integrations by parts allows to eliminate the unknown derivatives $(\mathbf{u}', p')$ from the expression (2). More precisely, multiplying the first equation in (1) by $\mathbf{v}_d$ and integrating by parts using Lemma 13 yields

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

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 (2.13) by $\mathbf{u}'$ and integrating by parts using Lemma 13, we obtain:

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

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

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

Now using the identity

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

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

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

Eventually, taking into account the boundary conditions in the systems (2.1), (2.11) and (1) yields:

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

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 $\operatorname{div}(\mathbf{u}) = 0$,

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

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

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

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

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

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

B Practical implementation of the shape optimization algorithm

In this section, we describe the practical code used in the numerical experiments of Section 2.5. In addition to the detailed comments accompanying the sources, we focus our discussion on the parts that should be modified for the user to implement a different geometric

or physical situation (i.e. to change the initial shape, the objective function, the shape derivative, etc.).

The numerical developments proposed in this work rely on the `FreeFem++` [80] 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.

B.1 Organization of the repository and of the program

The code may be downloaded from the address:

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

The main repository is organized as follows:

- The folder `./meshes` contains the mesh files associated to the initial shapes of the test cases of Section 2.5: `mesh1.mesh`, `mesh2.mesh`, etc.
- The `FreeFem++` source code used to generate these meshes is in the file `geometry.edp`.
- As the name suggests, the file `main.edp` contains the main routines of the optimization process.
- The file `macros.edp` contains several useful macros; see Section B.3.
- The file `curvature.edp` gathers the routines involved in the calculation of the mean curvature κ of shapes; see Section B.5.
- The files `run_case.sh` and `run_all.sh` are `shell` scripts containing the sample command lines needed to launch any, or all of the proposed test cases in Section 2.5.

B.2 Main parameters

The main program, written in the file `main.edp`, is executed by using the command line

```
FreeFem++ --param1 value1 ...  main.edp,
```

where `--param1`, ... are the computational parameters of the considered test case; see Table 1.

Seven geometric settings (associated to different meshes of the initial shape and applied boundary conditions) are implemented in our code, corresponding to values of the `config` parameter ranging from 1 to 7. The precise command lines used to launch these examples are supplied in the file `run_case.sh`. The first five configurations correspond to the numerical results of Section 2.5.

B.3 Main macros

Our program relies on macros insofar as possible: it is a convenient way in `FreeFem++` to ensure that the various operations carried out resemble their mathematical counterparts. The shortcuts that are consistently used throughout the implementation are stored in the file `macros.edp`; see Listing 1 for a sample.

```
/* Strain tensor */
macro EPS(u, v) [dx(u), 1./2*(dx(v)+dy(u)), 1./2*(dx(v)+dy(u)), dy(v)] // EOM

/* Jacobian matrix */
macro GRAD(u, v) [dx(u), dy(u), dx(v), dy(v)] // EOM

/* (u \cdot \nabla) V */
macro UgradV(u1,u2,v1,v2) [ [u1,u2]'*[dx(v1),dy(v1)] , [u1,u2]'*[dx(v2),dy(v2)] ]// EOM
```

Listing 1 – Several macros (from `macros.edp`)

B.4 Definition of the geometry and of the Finite Element setting

The meshes associated to the proposed test cases are supplied in the folder `./meshes`. The mesh `Th` corresponding to the considered situation (i.e. associated to the actual value of the `config` parameter) is read at the beginning of the `main.edp` file; see Listing 4.

```
/* Load initial mesh */
string meshname = "meshes/mesh"+config+".mesh";
```

`--config`	Number of the considered test-cases; `config` ranges from 1 to 7
`--navsto`	The Stokes (resp. Navier-Stokes) system models the flow if `navsto` is 0 (resp. 1)
`--tau`	Value of τ, initial step in the gradient descent, see (2.28)
`--errc`	Value of the stopping criterion $\varepsilon_{\text{stop}}$
`--gamma`	Value of the regularization parameter γ; see (2.31)
`--beta`	The constraint function $G(\Omega)$ is $\text{Vol}(\Omega)$ if `beta` is 1, and $\text{Per}(\Omega)$ if `beta` is 0.
`--delta`	The objective $J(\Omega)$ is the dissipated energy (2.3) if `delta` is 1, and the least-square discrepancy (2.4) if `delta` is 0.
`--binit`	Initial value for the penalty parameter `b` in (2.25)
`--btarget`	Limiting value for `b`
`--cv`	Desired constraint (volume or perimeter) over initial value for the constraint $G(\Omega)$
`--optraff`	0 for no remeshing, **1** for remeshing when necessary
`--raffinit`	Value of the `raff` parameter used in the routines for mesh adaptation (see Section B.8)

Table 1 – *Main parameters passed on the command line.*

```
cout << "Loading mesh " << meshname << "...";
Th = readmesh(meshname);
cout << "done." << endl;
cout.flush;
```

Listing 2 – Reading the initial shape (from `main.edp`)

The Finite Element spaces on the mesh `Th` are then defined as in Listing 3.

```
fespace Qh(Th,P1);
fespace Vh(Th,P2);

Vh    ux, uy, vx, vy, wx, wy, dux, duy, uxx, uyy, clx, cly;
Qh    p,q, mx, dpx, dpy, dp, qq, phix, phiy, kappa, phi, psi;
```

Listing 3 – Definition of the Finite Element spaces and functions (from `main.edp`)

These meshes may be generated using the code in the file `geometry.edp`, which can easily be modified and adapted to describe a different physical setting.

For instance, the code in Listing 4 allows to create the mesh of the initial shape in the bend test case of Section 2.5.1; see Figure 2.15 (top).

```
/* Bend with orthogonal inlet and outlet */
if (config==1) {
  border in(t=0,1){x=param(0,1./3,t);
                   y=0;
                   label=2;};
  border sig1(t=0,1){
           x=circlearcx(1,2./3,pi,pi/2,t);
           y=circlearcy(0,2./3,pi,pi/2,t);
           label=3;};
  border out(t=0,1){x=1;
                   y=param(2./3,1,t);
                   label=1;};
  border sig2(t=0,1){
           x=circlearcx(1,1,pi/2,pi,t);
           y=circlearcy(0,1,pi/2,pi,t);
           label=3;};
  Th=buildmesh(in(pp/2)+sig1(pp)
               +out(pp/2)+sig2(pp));
  Th=adaptmesh(Th,IsMetric=1,1./30);
}
```

Listing 4 – Creation of the initial mesh in the bend example of Section 2.5.1 (from `geometry.edp`)

B.5 Practical calculation of the mean curvature

The routines dedicated to the calculation of the mean curvature `kappa` of the boundary of the optimized shape are a little involved. They are gathered in the file
`curvature.edp` and in principle, they do not need to be modified.

The calculation of `kappa` in `main.edp` is then carried out along the lines of Listing 5.

```
kappa=0;
calculconnect(Th,ordre);
courbure(Th,ordre,kappa[]);
kappa=kc*kappa;
```

Listing 5 – Calculation of the mean curvature in `main.edp`

B.6 Resolution of the flow equations

As outlined in Section 2.4.2, the Navier-Stokes equations are solved iteratively thanks to the Newton method.

To achieve this, the Stokes equation is first defined as a variational problem; see Listing 6 and Remark 6:

```
problem stokes([ux, uy, p], [vx, vy, q]) =
  int2d(Th)(2*mu*tr(EPS(ux,uy))*EPS(vx,vy) - p * div(vx,vy))
  +int2d(Th)(div(ux,uy)*q)
  -int2d(Th)(p*q*epsilon)
  +on(3,ux=0,uy=0)
  +on(1,ux=clx, uy=cly);
```

Listing 6 – Variational problem for the Stokes system (from `main.edp`)

The Navier-Stokes system is solved for the velocity and pressure `[ux,uy,p]` by using the macro `ns` reprinted in Listing 7. In a nutshell, the Stokes system is solved as an initial guess; then, if the parameter `navsto` is set to 1, a loop is performed during which the Oseen equation is solved for the increment `[dux,duy,dp]`, from which `[ux,uy,p]` is updated.

```
  macro ns () {
/* Only solve when necessary if ns has
never been executed or if the mesh
has changed since the last resolution */
  if (solvefluid) {
/* Initialize Newton loop with the solution of Stokes */
```

```
7       stokes;
8   /* If we want to solve Navier-Stokes,
9   iterate successive Oseen problems */
10      if(navsto) {
11        int n;
12        real err=0;
13        cout << "Navier-Stokes";
14        /* Newton Loop  */
15        for(n=0; n< 15; n++) {
16    solve Oseen([dux,duy,dp],[vx,vy,qq]) =
17      int2d(Th)(2*nu*tr(EPS(dux,duy))*EPS(vx,vy)
18          + tr(UgradV(dux,duy, ux, uy))*[vx,vy]
19          + tr(UgradV(ux,uy,dux,duy))*[vx,vy]
20          - div(dux,duy)*qq - div(vx,vy)*dp
21          - epsilon*dp*qq)
22      +int2d(Th)(2*nu*tr(EPS(ux,uy))*EPS(vx,vy)
23           + tr(UgradV(ux,uy, ux, uy))*[vx,vy]
24           - div(ux,uy)*qq - div(vx,vy)*p
25           - epsilon*p*qq)
26      +on(1,3,dux=0,duy=0);
27
28    ux[] += dux[];
29    uy[] += duy[];
30    p[]  += dp[];
31    err = sqrt(int2d(Th)(tr(GRAD(dux,duy))*GRAD(dux,duy) + tr([dux,duy
          ])*[dux,duy]) / int2d(Th)(tr(GRAD(ux,uy))*GRAD(ux,uy) + tr([ux
          ,uy])*[ux,uy]));
32    cout << ".";
33    cout.flush;
34    if(err < arrns) break;
35        }
36        /* Newton loop has not converged */
37        if(err > arrns) {
38    cout << "NS Warning : non convergence : err = " << err << " / eps
          = " << epsilon << endl;
39        }
40      }
41      cout << endl;
42  /* It is not necessary to solve ns until the mesh is moved or
      adapted */
43      solvefluid = 0;
44      nflsolved++;
45    }
46  }//EOF
```

Listing 7 – Resolution of the flow equations (from `main.edp`)

B.7 Calculation of the objective function and of the shape derivative

The considered objective function $J(\Omega)$ in (2.2) is the energy dissipation (2.3) if the parameter `delta` is set to 1, and a least-square difference (2.4) between the fluid velocity and a target velocity if `delta` is 0. These are calculated from the macro in Listing 8.

```
/* Objective function = weighted sum of energy dissipation and least
   -square difference with a
 prescribed flow */
macro J() (2*delta*mu*int2d(Th)(tr(EPS(ux,uy))*EPS(ux,uy)) + ((1.-
   delta)/2)*int1d(Th,2)((ux-uxx)^2+(uy-uyy)^2)) //EOM
```

Listing 8 – Macro for the objective function (from `macros.edp`)

Likewise, the constraint function $G(\Omega)$ is $\mathrm{Vol}(\Omega)$ if `beta` is 1, and $\mathrm{Per}(\Omega)$ if `beta` is 0; these are calculated from the macro in Listing 9.

```
/* Constraint function = weighted sum of volume and perimeter */
macro contr(Th) (beta*int2d(Th)(1.)
                 +(1.-beta)*int1d(Th)(1.)) //EOM
```

Listing 9 – Macro for the constraint function (from `macros.edp`)

Thence, the value of the augmented Lagrangian functional is calculated by means of the macro `EL`, reprinted in Listing 10.

```
/* Augmented Lagrangian */
macro EL() (J/J0 + l*(contr(Th) - ctarget)/c0 + b/2 * ((contr(Th) -
   ctarget)^2)/(c0^2)) //EOM
```

Listing 10 – Macro for the augmented Lagrangian (from `macros.edp`)

At each iteration of the optimization loop (see Section B.8 below), the adjoint states `[vx,vy,q]` are calculated as the solution to (2.11) if `delta` is 1 or (2.13) if `delta` is 0. This is achieved by calling the macro `adjoint` reprinted in Listing 11. Notice the presence of the `navsto` variable in the variational problem for the adjoint states, corresponding to the term induced by the non linearity of the flow equation (2.1) if `navsto` equals 1.

```
macro adjoint() {
  solve probadjoint([vx, vy, q], [wx, wy, qq]) =
  int2d(Th) (2*nu*tr(EPS(vx, vy))*EPS(wx, wy)
         -q*div(wx, wy)
         -qq*div(vx, vy)
```

```
6           +navsto*(tr(UgradV(wx, wy, ux, uy))*[vx, vy]+tr(UgradV(ux,
                uy,wx,wy))*[vx,vy]))
7  +int2d(Th)(-4*nu*delta*tr(EPS(ux,uy))*EPS(wx,wy))
8  +int1d(Th,2)(-(1-delta)*((ux-uxx)*wx+(uy-uyy)*wy))
9  +on(1,3,5, vx=0, vy=0);
10 }//EOM
```

Listing 11 – Macro for the resolution of the adjoint system (from `main.edp`)

The shape derivatives of the considered objective and constraint functions $J(\Omega)$ and $G(\Omega)$, and that of the augmented Lagrangian $\mathcal{L}(\Omega, \ell, b)$ are then computed, again, thanks to a set of macros defined in the file `macros.edp`; see Listing 12.

```
1  /* Strain tensor */
2  macro EPS(u, v)
3    [dx(u), 1./2*(dx(v)+dy(u)),
4     1./2*(dx(v)+dy(u)), dy(v)] // EOM
5  /* Shape derivative of the objective function */
6  macro IJ()
7    (-2*delta*nu*tr(EPS(ux,uy))*EPS(ux,uy)
8    +2*nu*tr(EPS(ux,uy))*EPS(vx,vy)) //EOM
9  /* Shape gradient of the constraint function */
10 macro gradC() (beta*1+(1.-beta)*kappa) //EOM
11 /* Shape-gradient of the Lagrangian */
12 macro gradDF()
13   (IJ/J0 + l*gradC/c0
14   +b*gradC*(contr(Th)-ctarget)/(c0^2))//EOM
```

Listing 12 – Macros for shape derivatives (from `macros.edp`)

The shape gradient of the augmented Lagrangian on Γ is then extended to the whole computational mesh using the `regulbord` macro; the result of which is stored in the variable `[dpx, dpy]`; see Listing 13.

```
1  macro regulbord() {
2  solve regb([dpx, dpy],[phix, phiy]) =
3    int2d(Th)(gamma*tr(SIG(dpx, dpy))*EPS(phix, phiy))
4    +int1d(Th,3)(gamma1*tr(gradT(dpx))*gradT(phix))
5    +int1d(Th,3)(gamma1*tr(gradT(dpy))*gradT(phiy))
6    +int1d(Th,3)(gradDF*dotN(phix, phiy))
7    +int1d(Th,4)(1./epspen*dotN(dpx,dpy)*dotN(phix,phiy))
8    +on(1, 2, dpx=0)
9    +on(1, 2, dpy=0);
10 }//EOM
```

Listing 13 – Macro for the extension-regularization procedure of the shape gradient (from `main.edp`)

B.8 Main optimization loop : gradient descent with line search

Last but not least, we now discuss our implementation of the algorithm of Section 2.4.7 for the resolution of the shape optimization problem (2.2), properly speaking.

This is achieved by means of two nested loops; see Listings 14 and 15. The main, outermost loop, reprinted in Listing 14, drives the update of the shape. At the beginning of each iteration, the actual shape is stored in the mesh `Th2`; then the direct and adjoint problems are solved thanks to the macros `ns` and `adjoint` respectively (see Line 4); a descent direction `[dpx,dpy]` from the actual shape `Th2` is inferred by using the macro `regulbord` (Line 8). Meanwhile, the performance `L0` - i.e. the value of the augmented Lagrangian - of `Th2` is calculated (Line 9).

Then, starting from the input parameter `tau` an appropriate value of the time step `tau1` is found by the inner loop of Listing 15, which is described below.

This inner loop results in a mesh `Th` of the new shape; the coefficients ℓ^n and b^n of the augmented Lagrangian are eventually updated (Line 29). This main loop stops if either the maximum number of iterations `jjmax` is reached or if the ending criterion

$$\texttt{sv} \leq \texttt{errc}$$

is fulfilled; see Line 23 in Listing 14 and Section 2.4.7.

```
1  /* The algorithm stops when jj reaches jjmax or the ending criterion
       is low enough */
2  for (jj = 0; (sv > errc) && (jj < jjmax); jj++) {
3    Th2 = Th; // Keep a copy of the mesh
4    ns; // Solve the NS equation if needed
5    adjoint;  // Solve the adjoint system
6    /* Solve the velocity extension/regularization problem to get the
         descent direction;
7       the descent direction is [dpx,dpy] */
8    regulbord;
9    L0 = EL;  // Value of the augmented Lagrangian
10   tau1 = tau;
11
12   /* Inner loop for line search */
13    ****************************************
```

```
14    ** Linear seach loop: see Listing 15 below **
15    ****************************************
16
17    /* Maximum number of iterations has been reached, and no decrease
          in the value of the augmented Lagrangian is observed */
18    if (kk == kkmax) {
19      cout << "Warning : L_{n+1}>L_{n} (L0 = " << L0 << ", l = " << l
            << ")" << endl;
20    }
21
22    /* L^2 norm of the shape gradient, used as the ending criterion */
23    sv = sqrt(int1d(Th,3)(dpx^2+dpy^2));
24
25    /* Print output */
26    r << J << " " <<  EL << " " << contr(Th) << " " << l << " " << sv
          << " " << b << " " << minarea << endl;
27
28    /* Update of the values of the coefficients of the augmented
          Lagrangian */
29    l = l + b * (contr(Th) - ctarget);
30
31    /* Increase b if it is less than btarget */
32    if (b < btarget) {
33      b *= alpha;
34    }
35    cout << "jj = " << jj << endl;
36  }
```

Listing 14 – Main loop of the optimization algorithm (from `main.edp`)

Let us now describe the inner loop, which is nothing but a basic line search procedure for finding a suitable value of the time step `tau1`; see Listing 15. This procedure is initialized while `tau1=tau`, `tau` being a user-defined value. At each iteration of the inner loop, the current shape `Th2` is deformed along the descent direction `[dpx,dpy]` for a time `tau1`; this yields a new, 'attempt' mesh `Th` (Lines 6-30).

This shape `Th` is then evaluated: the flow equations are solved on `Th` (see Line 37), and the value `L1` of the augmented Lagrangian associated to `Th` is calculated (Line 40). If `L1` is smaller than the value `L0` of the augmented Lagrangian of the current shape `Th2`, the loop ends, and the 'attempt' mesh `Th` is accepted as the updated shape. Otherwise, the procedure is repeated from the beginning once the value of `tau1` has been divided by 2.

Note that, if after `kkmax=10` iterations of the line search proce-

dure, none of the produced 'attempt' meshes `Th` has produced a value `L1` of the augmented Lagrangian smaller than `L0`, the last iteration `kk = kkmax` is accepted nevertheless; the step used in this case being $\tau/2^{10}$, `Th` is then very close to `Th2`.

```
for (kk = 0;kk < kkmax; kk++) {
  cout << "movemesh tau = "<< tau1 << endl;
  minarea = checkmovemesh(Th2, [x + tau1*dpx, y + tau1*dpy]);

  /* Try to adapt the mesh in case one of the triangles becomes
     degenerate */
  if(optraff) {
    /* No adaptation if the minimal area is larger than parameter
       minarea0 or if we already tried remeshing 3 times in this
       loop */
    if (minarea > minarea0 || adaptcount>=3) {
      Th = movemesh(Th2, [x + tau1*dpx, y + tau1*dpy]);
      solvefluid = 1;
     }
     else {
      cout << "*** ADAPTMESH *** minarea = " << minarea << "
          minarea0 = " << minarea0 << endl;
      Th = adaptmesh(Th, hmax=raff, hmin=raff/sqrt(2), ratio=1.5);
      cout << " new minarea = " << minarea << endl;
      minarea = checkmovemesh(Th2, [x + tau1*dpx, y + tau1*dpy]);
      solvefluid = 1;
      kappa = 0;
      calculconnect(Th, ordre);
      adaptcount++;
    }

    if(adaptcount>=3) {
      cout << "Too many consecutive mesh adaptations. Giving up mesh
          adaptation" << endl;
    }
  }
  else {
    Th = movemesh(Th2, [x + tau1*dpx, y + tau1*dpy]);
    solvefluid = 1;
  }

  /* Calculate the mean curvature of the new shape */
  courbure(Th, ordre, kappa[]);
  kappa = kc * kappa;

  /* Solve the Navier-Stokes and adjoint equations on the new shape
     */
  ns;

  /* New value of the objective function */
  L1 = EL;
```

```
41   tau1/= 2;
42   cout << "L = " << L1 << " / L0 = " << L0 << " (variation = " <<
         100*(L1-L0)/L0 << "%)" << endl;
43
44   /* Accept iteration as soon as the value of the augmented
         Lagrangian is decreased */
45   if (L1 < L0) {
46     break;
47   }
48 }
```

Listing 15 – Line search in the optimization algorithm (from `main.edp`)

One final word about remeshing is in order. If the `optraff` input parameter equals 1, then whenever the mesh of the shape is deformed, the minimum area of an element in the tentative mesh `Th` is compared to the parameter `minarea0` (see Lines 6-30 in Listing 15). If smaller, the mesh is adapted thanks to the `adaptmesh` command of `FreeFem++` (Line 14 in Listing 15). The resulting mesh has edges with length comprised betwen `raff` and `raff/sqrt(2)`, where `raff` stems from the `raffinit` parameter from the command line.

Note that, if mesh adaptation occurs, the connectivity of the boundary has to be calculated anew by calling `calculconnect` (viz. line 33), so that the routines described in Section B.5 may be used to calculate the curvature of the shape.

C Proof of the axisymmetry result on a fixed domain

This section is dedicated to the proof of Theorem 11 as well as some related remarks. Lemma 14 is the main ingredient for the proof, Lemmas 15 and 16 are technical lemmas, postponed at the end of this Appendix.

Lemma 14. *Let* $\mathbf{v} = R_{-\eta}\mathbf{u} \circ R_\eta$. *Then*

$$\begin{aligned} \mathcal{E}(\mathbf{v}) &= \mathcal{E}(\mathbf{u}) \\ \int_{\Gamma_{in}} \mathbf{v} \cdot n \, ds &= \int_{\Gamma_{in}} \mathbf{u} \cdot n \, ds = Q \\ \operatorname{div}(\mathbf{v}) &= \operatorname{div}(\mathbf{u}) = 0 \end{aligned} \tag{11}$$

Proof. The second equality of (11) is proven using a change of variables on Γ_{out}, while the third is proven using Lemma 15. As for $\mathcal{E}(\mathbf{v})$, a substitution both on Ω and Γ yields

$$\begin{aligned}
\mathcal{E}(\mathbf{v}) &= \int_\Omega e(\mathbf{v}) : e(\mathbf{v}) \, \mathrm{d}x + \beta \int_\Gamma |\mathbf{v}|^2 \, \mathrm{d}s \\
&= \int_\Omega R_{-\eta}(e(\mathbf{u}) \circ R_\eta) R_\eta : R_{-\eta}(e(\mathbf{u}) \circ R_\eta) R_\eta \, \mathrm{d}x + \beta \int_\Gamma |\mathbf{u} \circ R_\eta|^2 \, \mathrm{d}s \\
&= \int_\Omega (e(\mathbf{u}) \circ R_\eta) : (e(\mathbf{u}) \circ R_\eta) \, \mathrm{d}x + \beta \int_\Gamma |\mathbf{u} \circ R_\eta|^2 \, \mathrm{d}s \\
&= \int_{R_\eta \Omega} e(\mathbf{u}) : e(\mathbf{u}) \operatorname{Jac}(T)^{-1} \, \mathrm{d}x + \beta \int_{R_\eta \Gamma} |\mathbf{u}|^2 \operatorname{Jac}_\Gamma(T)^{-1} \, \mathrm{d}s
\end{aligned} \tag{12}$$

Lemma 16 was used to expand $e(\mathbf{v})$. Since R_η is an orthogonal transformation, $\operatorname{Jac}(R_\eta) = |\det(\nabla R_\eta)| = 1$. Let us recall the definition of $\operatorname{Jac}_\Gamma(R_\eta)$ (see [81, Chapter 5])

$$\operatorname{Jac}_\Gamma(R_\eta) = |R_\eta^{-T} n| \operatorname{Jac}(R_\eta)$$

Again, since $R_\eta^{-T} = R_\eta$ is an orthogonal transformation, it preserves the norm of the normal vector $|n| = 1$. Finally, Ω and Γ are preserved by R_η, hence

$$\mathcal{E}(\mathbf{v}) = \mathcal{E}(\mathbf{u})$$

□

Proof. of Theorem 11

From Lemma 14, if $\mathbf{u}$ is a solution of problem (3.51), then so is $\mathbf{v} = R_{-\eta}\mathbf{u} \circ R_\eta$. As proven in Proposition 5, problem (3.51) has a unique solution in $H^1(\Omega)^d$. Therefore $\mathbf{v} = \mathbf{u}$ for all $\eta \in \mathbb{R}$.

Without any assumption on $\mathbf{u}$, $\nabla \mathbf{u}$ writes in cylindrical coordinates

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_r}{\partial_r} & \frac{1}{r}\frac{\partial u_r}{\partial_\theta} - \frac{u_\theta}{r} & \frac{\partial u_r}{\partial_z} \\ \frac{\partial u_\theta}{\partial_r} & \frac{1}{r}\frac{\partial u_\theta}{\partial_\theta} + \frac{u_r}{r} & \frac{\partial u_\theta}{\partial_z} \\ \frac{\partial u_z}{\partial_r} & \frac{1}{r}\frac{\partial u_z}{\partial_\theta} & \frac{\partial u_z}{\partial_z} \end{pmatrix}$$

Note that this tensor is expressed in basis (e_r, e_θ, e_z). This means that for example,

$$\nabla u e_r \cdot e_\theta = \frac{\partial u_\theta}{\partial r}$$

APPENDIX . PROOF OF THE AXISYMMETRY RESULT ON A FIXED DOMAIN

The main difference from Cartesian coordinates is that since (e_r, e_θ, e_z) is a moving basis, additional terms appear, therefore in general for $i, j = r, \theta, z$:

$$\nabla u e_i \cdot e_j \neq \frac{\partial u_j}{\partial x_i}$$

Matrix transposition commutes with any orthogonal change of coordinates, allowing us to write

$$\begin{aligned} e(\mathbf{u}) &= \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \\ &= \begin{pmatrix} \frac{\partial u_r}{\partial_r} & \frac{1}{2}\left(\frac{1}{r}\frac{\partial u_r}{\partial_\theta} - \frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial_r}\right) & \frac{1}{2}\left(\frac{\partial u_z}{\partial_r} + \frac{\partial u_r}{\partial_z}\right) \\ \frac{1}{2}\left(\frac{1}{r}\frac{\partial u_r}{\partial_\theta} - \frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial_r}\right) & \frac{1}{r}\frac{\partial u_\theta}{\partial_\theta} + \frac{u_r}{r} & \frac{1}{2}\left(\frac{\partial u_\theta}{\partial_z} + \frac{1}{r}\frac{\partial u_z}{\partial_\theta}\right) \\ \frac{1}{2}\left(\frac{\partial u_z}{\partial_r} + \frac{\partial u_r}{\partial_z}\right) & \frac{1}{2}\left(\frac{\partial u_\theta}{\partial_z} + \frac{1}{r}\frac{\partial u_z}{\partial_\theta}\right) & \frac{\partial u_z}{\partial_z} \end{pmatrix} \end{aligned}$$

We proved that $\mathbf{u}$ does not depend on θ, so all derivatives w.r.t θ vanish.

$$e(\mathbf{u}) = \begin{pmatrix} \frac{\partial u_r}{\partial_r} & \frac{1}{2}\left(-\frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial_r}\right) & \frac{1}{2}\left(\frac{\partial u_z}{\partial_r} + \frac{\partial u_r}{\partial_z}\right) \\ \frac{1}{2}\left(-\frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial_r}\right) & \frac{u_r}{r} & \frac{1}{2}\frac{\partial u_\theta}{\partial_z} \\ \frac{1}{2}\left(\frac{\partial u_z}{\partial_r} + \frac{\partial u_r}{\partial_z}\right) & \frac{1}{2}\frac{\partial u_\theta}{\partial_z} & \frac{\partial u_z}{\partial_z} \end{pmatrix}$$

Finally, the Frobenius product is invariant through an orthogonal change of basis[1]. Hence $\mathcal{E}(\mathbf{u})$ can be decomposed as a sum of terms dependent on (u_r, u_z) on the one hand and u_θ on the other hand.

$$\mathcal{E}(\mathbf{u}) = \mathcal{E}_1(u_r, u_z) + \mathcal{E}_2(u_\theta)$$

with :

$$\begin{aligned} \mathcal{E}_1(u_r, u_z) &= \nu \int_\Omega \left(\frac{\partial u_r}{\partial r}\right)^2 + \left(\frac{u_r}{r}\right)^2 + \left(\frac{\partial u_z}{\partial z}\right)^2 + \frac{1}{2}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right)^2 r \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}z \\ &\quad + \frac{\beta}{2}\int_\Gamma (u_r^2 + u_z^2)\, r \,\mathrm{d}\theta \,\mathrm{d}z \\ \mathcal{E}_2(u_\theta) &= \frac{\nu}{2}\int_\Omega \left(-\frac{u_\theta}{r} + \frac{\partial u_\theta}{\partial r}\right)^2 + \left(\frac{\partial u_\theta}{\partial z}\right)^2 r \,\mathrm{d}r \,\mathrm{d}\theta \,\mathrm{d}z + \frac{\beta}{2}\int_\Gamma u_\theta^2 \, r \,\mathrm{d}\theta \,\mathrm{d}z \end{aligned}$$

[1] $R^T A R : R^T B R = \mathrm{tr}((R^T A R)^T R^T B R) = \mathrm{tr}(R^T A^T R R^T B R) = \mathrm{tr}(R R^T A^T B) = A : B$

The constraint involved in problem (3.51) can be re-cast as $C(u_r, u_z) = 0$ with

$$C(u_r, u_z) = \left(\mathbf{u} \cdot n, \int_{\Gamma_{\text{in}}} \mathbf{u} \cdot n \, \mathrm{d}s, \operatorname{div}(\mathbf{u})\right) \in H^{\frac{1}{2}}(\Gamma) \times R \times L^2(\Omega)$$

While it is clear that neither $\mathbf{u} \cdot n$ nor $\int_{\Gamma_{\text{in}}} \mathbf{u} \cdot n \, \mathrm{d}s$ depend on $\mathbf{u}_\theta$, it is also the case for $\operatorname{div}(\mathbf{u})$ since the third term vanishes in

$$\operatorname{div}(\mathbf{u}) = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}$$

With these considerations in mind, it is possible to re-write problem (3.51)

$$\min_{\mathbf{u} \in H^1(\Omega), C(u_r, u_z) = 0} \mathcal{E}_1(u_r, u_\theta) + \mathcal{E}_2(u_\theta)$$

Since variables (u_r, u_z) and u_θ are independent, this problem can be split in two independent sub-problems that are equivalent to (3.51)

$$\begin{aligned} \min_{(u_r, u_z)} \quad & \mathcal{E}_1(u_r, u_\theta) \\ C(u_r, u_z) = 0 & \end{aligned} \tag{13}$$

and

$$\min_{\mathbf{u}_\theta} \mathcal{E}_2(u_\theta) \tag{14}$$

Since $u_\theta \equiv 0$ is the unique solution of problem (14), the solution of the original problem (3.51) also satisfies this property. □

Lemma 15. *The following holds*

$$\operatorname{div}(\mathbf{v}) = \operatorname{div}(\mathbf{u}) \circ R_\eta \tag{15}$$

Proof. Using (16), and the fact that for square matrices A and B, $\operatorname{tr}(AB) = \operatorname{tr}(BA)$,

$$\begin{aligned} \operatorname{div}(\mathbf{v}) &= \operatorname{tr}(\nabla \mathbf{v}) \\ &= \operatorname{tr}(R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta) \\ &= \operatorname{tr}(R_\eta R_{-\eta}(\nabla \mathbf{u} \circ R_\eta)) \\ &= \operatorname{tr}(\nabla \mathbf{u} \circ R_\eta) \\ &= \operatorname{div}(\mathbf{u}) \circ R_\eta \end{aligned}$$

□

Lemma 16. *Let* $\mathbf{v} = R_{-\eta} \circ \mathbf{u} \circ R_\eta$. *The the following relationships hold :*

$$\nabla \mathbf{v} = R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta \tag{16}$$

$$e(\mathbf{v}) = R_{-\eta}(e(\mathbf{u}) \circ R_\eta) R_\eta \tag{17}$$

Proof. Using the chain rule on Jacobian matrices, and the fact that $R_{-\eta}$ and R_η are linear transformations,

$$\begin{aligned}
\nabla(R_{-\eta} \circ \mathbf{u} \circ R_\eta) &= \nabla(R_{-\eta} \circ (\mathbf{u} \circ R_\eta)) \\
&= \nabla R_{-\eta} \circ (\mathbf{u} \circ R_\eta) \nabla(\mathbf{u} \circ R_\eta) \\
&= R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) \nabla R_\eta \\
&= R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta
\end{aligned}$$

The corresponding relation holds for the symmetric part of $\nabla \mathbf{v}$

$$\begin{aligned}
e(\mathbf{v}) &= \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \\
&= \frac{1}{2}(R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta + (R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta)^T) \\
&= \frac{1}{2}(R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta + (R_\eta^T (\nabla \mathbf{u} \circ R_\eta)^T R_{-\eta}^T)) \\
&= \frac{1}{2}(R_{-\eta}(\nabla \mathbf{u} \circ R_\eta) R_\eta + (R_{-\eta}(\nabla \mathbf{u} \circ R_\eta)^T R_\eta)) \\
&= R_{-\eta}(e(\mathbf{u}) \circ R_\eta) R_\eta
\end{aligned}$$

□

Remark 12. *We have used a variational formalism to prove that* $\mathbf{u}$ *is axisymmetric (Theorem 11). For illustration purposes, we show that the strong writing of the boundary conditions is preserved by rotation. Indeed, the following equalities hold on* Γ *for any vector field* $\mathbf{u}$ *and* $\mathbf{v} = R_{-\eta} \circ \mathbf{u} \circ R_\eta$ *:*

$$\begin{cases} [e(\mathbf{v})\mathbf{n}_\eta + \beta \mathbf{v}]_{\tau_\eta} = R_{-\eta}[e(\mathbf{u})\mathbf{n} + \beta \mathbf{u}]_\tau \circ R_\eta \\ \mathbf{v} \cdot n_\eta = (\mathbf{u} \cdot \mathbf{n}) \circ R_\eta. \end{cases} \tag{18}$$

where $\mathbf{n}_\eta = R_{-\eta} \circ \mathbf{n} \circ R_\eta$ *and*

$$[\mathbf{F}]_{\tau_\eta} = \mathbf{F} - (\mathbf{F} \cdot \mathbf{n}_\eta)\mathbf{n}_\eta$$

Proof. From Lemma 16,

$$e(\mathbf{v})\mathbf{n}_\eta + \beta\mathbf{v} = R_{-\eta}(e(\mathbf{u})\mathbf{n} + \beta\mathbf{u}) \circ R_\eta$$

for any vectorial field $\mathbf{w}$ defined over $\partial\Omega$,

$$[R_{-\eta}\mathbf{w} \circ R_\eta]_{\tau_\eta} = R_{-\eta}[\mathbf{w}]_\tau \circ R_\eta$$

Taking $\mathbf{w} = e(\mathbf{u})\mathbf{n} + \beta\mathbf{u}$ proves (18). As for the non-penetration condition,

$$\begin{aligned}
\mathbf{v} \cdot \mathbf{n}_\eta &= R_{-\eta}(\mathbf{u} \circ R_\eta) \cdot R_{-\eta}\mathbf{n} \circ R_\eta \\
&= R_{-\eta}^T R_{-\eta}(\mathbf{u} \circ R_\eta) \cdot \mathbf{n} \circ R_\eta \\
&= (\mathbf{u} \cdot \mathbf{n}) \circ R_\eta
\end{aligned}$$

□

Bibliography

[1] N. Aage, T. H. Poulsen, A. Gersborg-Hansen, and O. Sigmund. Topology optimization of large scale stokes flow problems. *Struct. Multidisc. Optim.*, 35:175–180, 2008.

[2] E. Abbena, S. Salamon, and A. Gray. *Modern differential geometry of curves and surfaces with Mathematica*. Chapman and Hall/CRC, 2017.

[3] 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.

[4] L. Afraites, M. Dambrine, and D. Kateb. On second order shape optimization methods for electrical impedance tomography. *SIAM Journal on Control and Optimization*, 47(3):1556–1590, 2008.

[5] V. Agoshkov, A. Quarteroni, and G. Rozza. Shape design in aorto-coronaric bypass anastomoses using perturbation theory. *SIAM Journal on Numerical Analysis*, 44(1):367–384, 2006.

[6] G. Allaire. Homogenization of the navier-stokes equations in open sets perforated with tiny holes i. abstract framework, a volume distribution of holes. *Archive for Rational Mechanics and Analysis*, 113(3):209–259, 1991.

[7] G. Allaire. Homogenization of the navier-stokes equations with a slip boundary condition. *Communications on pure and applied mathematics*, 44(6):605–641, 1991.

[8] G. Allaire. *Conception optimale de structures*, volume 58. Springer, 2007.

[9] G. Allaire. *Conception optimale de structures*, volume 58 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, 2007. With the collaboration of Marc Schoenauer (INRIA) in the writing of Chapter 8.

[10] G. Allaire. *Shape optimization by the homogenization method*, volume 146. Springer Science & Business Media, 2012.

[11] G. Allaire, F. De Gournay, F. Jouve, and A.-M. Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and cybernetics*, 34(1):59, 2005.

[12] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics*, 194(1):363–393, 2004.

[13] G. Allaire and O. Pantz. Structural optimization with `freefem++`. *Structural and Multidisciplinary Optimization*, 32(3):173–181, 2006.

[14] S. Amstutz. The topological asymptotic for the navier-stokes equations. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(3):401–425, 2005.

[15] B. Aoubiza, J. M. Crolet, and A. Meunier. On the mechanical characterization of compact bone structure using the homogenization theory. *Journal of Biomechanics*, 29(12):1539–1547, 1996.

[16] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients ? *SIAM Journal on Applied Mathematics*, 63(6):2128–2154, 2003.

[17] M. Badra, F. Caubet, and M. Dambrine. Detecting an obstacle immersed in a fluid by shape optimization methods. *Math. Models Methods Appl. Sci.*, 2011.

[18] T. J. Baker. Mesh movement and metamorphosis. *Eng. Comput.*, 18(1):188–198, 2002.

[19] C. Belin, L. Joly, and F. Detcheverry. Optimal shape of entrances for a frictionless nanochannel. *Physical Review Fluids*, 1(5):054103, 2016.

[20] J. A. Bello, E. Fernandez-Cara, J. Lemoine, and J. Simon. The differentiability of the drag with respect to the variations of a lipschitz domain in a navier–stokes flow. *SIAM Journal on Control and Optimization*, 35(2):626–640, 1997.

[21] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, 71(2):197–224, 1988.

[22] M. P. Bendsoe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[23] G. Benga. The first discovered water channel protein, later called aquaporin 1: molecular characteristics, functions and medical implications. *Molecular aspects of medicine*, 33(5):518–534, 2012.

[24] M. Bergounioux and Y. Privat. Shape optimization with Stokes constraints over the set of axisymmetric domains. *SIAM J. Control Optim.*, 51(1):599–628, 2013.

[25] C. Bernardi, M. Dauge, Y. Maday, and M. Azaïez. *Spectral methods for axisymmetric domains*, volume 3. Gauthier-Villars Paris, 1999.

[26] L. Bocquet and J.-L. Barrat. Flow boundary conditions from nano- to micro-scales. *Soft Matter*, 3:685–693, 2007.

[27] L. Bocquet and J.-L. Barrat. Flow boundary conditions from nano- to micro-scales. *Soft Matter*, 3:685–693, 2007.

[28] M. Bonnivard. On the stability of self-propelled bodies with respect to their shape motion. *Mathematical Models and Methods in Applied Sciences*, 21(04):667–691, 2011.

[29] M. Bonnivard, F. Omnès, and Y. Privat. Modeling and optimization of hourglass-shaped aquaporins. *Mathematical Models and Methods in Applied Sciences*, 0(0):1–36, 0.

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

[31] J.-M. Bourot. On the numerical computation of the optimum profile in stokes flow. *Journal of Fluid Mechanics*, 65(3):513–515, 1974.

[32] R. P. Brent. *Algorithms for minimization without derivatives*. Prentice Hall, 1973.

[33] H. Brezis, A. Ambrosetti, S. I. S. di Studi Avanzati, T. A. Bahri, F. Browder, L. Caffarelli, L. C. Evans, M. Giaquinta, D. Kinderlehrer, S. Klainerman, et al. Progress in nonlinear differential equations and their applications. 2005.

[34] C.-H. Bruneau, F. Chantalat, A. Iollo, B. Jordi, and I. Mortazavi. Modelling and shape optimization of an actuator. *Structural and Multidisciplinary Optimization*, 48(6):1143–1151, 2013.

[35] 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.

[36] D. Bucur and A. Giacomini. Shape optimization problems with Robin conditions on the free boundary. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 33(6):1539–1568, 2016.

[37] D. Bucur, A. Giacomini, and P. Trebeschi. The Robin–Laplacian problem on varying domains. *Calc. Var. Partial Differential Equations*, 55(6):55:133, 2016.

[38] M. Burger. A framework for the construction of level set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*, 5(3):301–329, 2003.

[39] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1):344–362, 2004.

[40] J. Burkardt, M. Gunzburger, and J. Peterson. Insensitive functionals, inconsistent gradients, spurious minima, and regularized functionals in flow optimization problems. *International Journal of Computational Fluid Dynamics*, 16(3):171–185, 2002.

[41] J. M. Carbrey and P. Agre. Discovery of the aquaporins and development of the field. In *Aquaporins*, pages 3–28. Springer, 2009.

[42] 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.

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

[44] V. J. Challis and J. K. Guest. Level set topology optimization of fluids in stokes flow. *International journal for numerical methods in engineering*, 79(10):1284–1308, 2009.

[45] S. Chandrashekar and K. P. Chen. Minimizing hydraulic resistance of a plant root by shape optimization. *Mechanics Research Communications*, 75:44 – 48, 2016.

[46] D. Chenais. On the existence of a solution in a domain identification problem. *J. Math. Anal. Appl.*, 52(2):189–219, 1975.

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

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

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

[50] C. Dapogny, P. Frey, F. Omnès, and Y. Privat. Geometrical shape optimization in fluid mechanics using freefem++. *Structural and Multidisciplinary Optimization*, Jun 2018.

[51] F. De Gournay. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. *SIAM journal on control and optimization*, 45(1):343–367, 2006.

[52] 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(3):767–799, 2011.

[53] 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(3):767–799, 2011.

[54] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries*, volume 22 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2011. Metrics, analysis, differential calculus, and optimization.

[55] B. V. der Bruggen and C. Vandecasteele. Removal of pollutants from surface water and groundwater by nanofiltration: overview of possible applications in the drinking water industry. *Environmental Pollution*, 122(3):435 – 445, 2003.

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

[57] 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(37):3898–3914, 2007.

[58] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003.

[59] O. Dorn and D. Lesselier. Level set methods for inverse scattering. *Inverse Problems*, 22(4):R67, 2006.

[60] 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*, 222:487–499, 2008.

[61] P. Dubois, C. Dedeban, and J. P. Zolesio. Antenna's shape optimization and reconstruction by level-set 3d. In *2006 IEEE Antennas and Propagation Society International Symposium*, pages 581–584, July 2006.

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

[63] A. Evgrafov. Topology optimization of slightly compressible fluids. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 86(1):46–62, 2006.

[64] L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni. Numerical Treatment of Defective Boundary Conditions for the Navier-Stokes Equations. *SIAM Journal on Numerical Analysis*, 40(1):376–401, 2002.

[65] M. Fortin and R. Glowinski. *Augmented Lagrangian methods*, volume 15 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, 1983. Applications to the numerical solution of boundary value problems, Translated from the French by B. Hunt and D. C. Spicer.

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

[67] C. Fritzmann, J. Löwenberg, T. Wintgens, and T. Melin. State-of-the-art of reverse osmosis desalination. *Desalination*, 216(1):1 – 76, 2007.

[68] 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(4):A1846–A1871, 2015.

[69] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber. Topology optimization of channel flow problems. *Structural and Multidisciplinary Optimization*, 30(3):181–192, 2005.

[70] K. C. Giannakoglou and D. I. Papadimitriou. *Adjoint methods for shape optimization*. Springer, 2008.

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

[72] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations*, volume 5 of *Springer series in computational mathematics*. Springer Verlag, 1986.

[73] S. Gravelle, L. Joly, F. Detcheverry, C. Ybert, C. Cottin-Bizonne, and L. Bocquet. Optimizing water permeability

through the hourglass shape of aquaporins. *Proceedings of the National Academy of Sciences*, 110(41):16367–16372, 2013.

[74] S. Gravelle, L. Joly, C. Ybert, and L. Bocquet. Large permeabilities of hourglass nanopores: From hydrodynamics to single file transport. *The Journal of chemical physics*, 141(18):18C526, 2014.

[75] P. Gravesen, J. Branebjerg, and J. O. S. Microfluidics-a review. *Journal of Micromechanics and Microengineering,*, 3(4), 1994.

[76] J. Guest and J. Prévost. Topology optimization of creeping fluid flows using a darcy-stokes finite element. *Int. J. Numer. Meth. Engng*, 66:461–484, 2006.

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

[78] M. D. Gunzburger and H. Kim. Existence of an optimal solution of a shape control problem for the stationary navier–stokes equations. *SIAM Journal on control and optimization*, 36(3):895–909, 1998.

[79] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.

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

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

[82] 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(19-20):1057–1061, 2008.

[83] A. Henrot and Y. Privat. What is the optimal shape of a pipe? *Archive for rational mechanics and analysis*, 196(1):281–302, 2010.

[84] G. Hetsroni, A. Mosyak, E. Pogrebnyak, and L. Yarin. Fluid flow in micro-channels. *International Journal of Heat and Mass Transfer*, 48(10):1982–1998, 2005.

[85] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 22(5):325–352, 1996.

[86] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

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

[88] G. Hummer, J. C. Rasaiah, and J. P. Noworyta. Water conduction through the hydrophobic channel of a carbon nanotube. *Nature*, 2001.

[89] K. Ito, K. Kunisch, and G. H. Peichl. Variational approach to shape derivatives. *ESAIM: Control, Optimisation and Calculus of Variations*, 14(3):517–539, 2008.

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

[91] S. K. Kannam, B. D. Todd, J. S. Hansen, and P. J. Daivis. How fast does water flow in carbon nanotubes? *The Journal of Chemical Physics*, 138(9):094701, 2013.

[92] G. Karniadakis, A. Beskok, and N. Aluru. *Simple Fluids in Nanochannels*. Springer, 2005.

[93] Y.-M. Koh. Vorticity and viscous dissipation in an incompressible flow. *KSME Journal*, 8(1):35–42, Mar 1994.

[94] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, 46(3):311–326, 2012.

[95] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalized shape optimization of fluids with the lattice boltzmann method. *International Journal for Numerical Methods in Fluids*, 65(5):496–519, 2011.

[96] S. Kreissl, G. Pingen, and K. Maute. Topology optimization for unsteady flow. *International Journal for Numerical Methods in Engineering*, 87(13):1229–1253, 2011.

[97] H. Lamb. *Hydrodynamics, 6th ed.* Dover Publications, 1945.

[98] P. Le Tallec and E. Laporte. *Numerical Methods in Sensitivity Analysis and Shape Optimization*. Springer Extra Materials, 2003.

[99] S. H. Lee, S. G. Hong, K. H. Lee, H. S. Choi, and I. H. Park. Shape optimization for maximizing ferromagnetic repulsive force using shape sensitivity and adaptive level set method. In *Electrical Machines and Systems (ICEMS), 2015 18th International Conference on*, pages 290–294, Oct 2015.

[100] R. D. Letterman, A. W. W. Association, et al. *Water quality and treatment*. McGraw-Hill„ 1999.

[101] Y. Li, Q. Wang, Y. Dai, H. Wang, and J. Wang. Shape optimization of ferromagnetic pole of a ferromagnetic-superconducting mri magnet. *IEEE Transactions on Applied Superconductivity*, 26(7):1–5, Oct 2016.

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

[103] A. Litman, D. Lesselier, and F. Santosa. Reconstruction of a two-dimensional binary obstacle by controlled evolution of a level-set. *Inverse Problems*, 14(3):685, 1998.

[104] Y. Liu, R. So, and C. Zhang. Modeling the bifurcating flow in a human lung airway. *Journal of Biomechanics*, 35(4):465 – 473, 2002.

[105] A. Marsden L., M. Wang, J. J. Dennis, and P. Moin. Suppression of vortex-shedding noise via derivative-free shape optimization. *Phys. Fluids, 16*, 2004.

[106] B. Mauroy, M. Filoche, E. Weibel, and B. Sapoval. An optimal bronchial tree may be dangerous. *Nature*, 427(6975):633, 2004.

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

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

[109] P. Morin, R. H. Nochetto, M. S. Pauletti, and M. Verani. Adaptive finite element method for shape optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 18(4):1122–1149, 2012.

[110] F. Murat and J. Simon. Sur le contrôle par un domaine géométrique. *Technical report RR-76005*, 1976.

[111] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[112] A. A. Novotny and J. Sokołowski. *Topological derivatives in shape optimization*. Springer Science & Business Media, 2012.

[113] A. Novruzi and M. Pierre. Structure of shape derivatives. *Journal of Evolution Equations*, 2(3):365–382, 2002.

[114] E. Orman and G. Durmuş. Comparison of shape optimization techniques coupled with genetic algorithm for a wind turbine airfoil. In *2016 IEEE Aerospace Conference*, pages 1–7, March 2016.

[115] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[116] H. G. Park and Y. Jung. Carbon nanofluidics of rapid water transport for energy applications. *Chem. Soc. Rev.*, 43(565), 2014.

[117] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde-based fast local level set method. *Journal of computational physics*, 155(2):410–438, 1999.

[118] J. E. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373 – 391, 2010.

[119] G. Pingen, A. Evgrafov, and K. Maute. Topology optimization of flow domains using the lattice boltzmann method. *Structural and Multidisciplinary Optimization*, 34(6):507–524, 2007.

[120] G. Pingen, M. Waidmann, A. Evgrafov, and K. Maute. A parametric level-set approach for topology optimization of flow domains. *Structural and Multidisciplinary Optimization*, 41(1):117–131, 2010.

[121] O. Pironneau. On optimum profiles in stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973.

[122] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(01):97–110, 1974.

[123] O. Pironneau. *Optimal shape design for elliptic systems*. Springer Science & Business Media, 2012.

[124] J. Reuther, J. J. Alonso, M. J. Rimlinger, and A. Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers & fluids*, 28(4-5):675–700, 1999.

[125] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In *34th Aerospace Sciences Meeting and Exhibit*, page 94, 1996.

[126] D. Richards and M. Amos. Shape optimization with surface-mapped cppns. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2016.

[127] S. Schmidt and V. Schulz. Shape derivatives for general objective functions and the incompressible navier-stokes equations. *Control and Cybernetics*, 39(3):677–713, 2010.

[128] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of computational physics*, 163(2):489–528, 2000.

[129] K. M. Shirvan, R. Ellahi, S. Mirzakhanlari, and M. Mamourian. Enhancement of heat transfer and heat exchanger effectiveness in a double pipe heat exchanger filled with porous media: Numerical simulation and sensitivity analysis of turbulent fluid flow. *Applied Thermal Engineering*, 109, Part A:761 – 774, 2016.

[130] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001.

[131] J. Sokołowski and J.-P. Zolésio. *Introduction to shape optimization*. Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992.

[132] K. Suzuki and N. Kikuchi. A homogenization method for shape and topology optimization. *Computer methods in applied mechanics and engineering*, 93(3):291–318, 1991.

[133] K. Takata, T. Matsuzaki, and Y. Tajika. Aquaporins: water channel proteins of the cell membrane. *Progress in Histochemistry and Cytochemistry*, 39(1):1 – 83, 2004.

[134] R. Talebitooti, M. Shojaeefard, and S. Yarmohammadisatri. Shape design optimization of cylindrical tank using b-spline curves. *Computers & Fluids*, 109:100 – 112, 2015.

[135] A. S. Verkman. Aquaporins: translating bench research to human disease. *Journal of Experimental Biology*, 212(11):1707–1715, 2009.

[136] L. Wang, Y. Fan, and L. Luo. Lattice boltzmann method for shape optimization of fluid distributor. *Computers & Fluids*, 94:49 – 57, 2014.

[137] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer methods in applied mechanics and engineering*, 192(1):227–246, 2003.

[138] X. Wang, Y. Mei, and M. Wang. Incorporating topological derivatives into level set methods for structural topology optimization. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4564, 2004.

[139] S. Zhou and Q. Li. A variational level set method for the topology optimization of steady-state navier-stokes flow. *J. Comput. Phys.*, 227:10178–10195, 2008.

[140] F. Zhu, E. Tajkhorshid, and K. Schulten. Theory and simulation of water permeation in aquaporin-1. *Biophysical Journal*, 86(1):50–57, 2004.

Optimisation geométrique appliquée à la mécanique des fluides incompressibles

Résumé :
Cette thèse de mathématiques appliquées est consacrée à la modélisation et à l'exploration de techniques numériques d'optimisation de la forme d'objets au contact de fluides.
Le premier chapitre est consacré à un algorithme d'optimisation géométrique mis en œuvre dans le logiciel `optiflow`, dans le cas où le bord à optimiser est associé à des conditions de non-glissement. L'implémentation est mise en ligne et accompagnée d'une notice d'utilisation. Il est ainsi possible de l'utiliser pour des applications de la vie réelle, par exemple pour l'optimisation de la géométrie d'un pipeline, de conduits de climatisation, etc.
Dans le second chapitre, nous décrivons une façon de modéliser l'écoulement fluide à travers une aquaporine. Après avoir précisé et motivé le modèle fluide, nous prouvons l'existence d'une forme optimale pour le critère d'énergie dissipée par le fluide. Les conditions de bord de glissement partiel font apparaître des difficultés dans le calcul de sensibilité, nous présentons un traitement numérique spécifique pour y remédier. Enfin, plusieurs exemples numériques sont présentés et commentés.

Mots-clés : Équations de Navier-Stokes, Équations de Stokes, Optimisation de formes, Lagrangien augmenté, Aquaporines

Geometry optimization applied to incompressible fluid mechanics

Abstract :
This applied mathematics thesis is dedicated to the modelling and exploration of numerical geometry optimization techniques.
The first chapter is dedicated to a geometry optimization algorithm implemented in `optiflow`, in the case where the boundary to optimize is associated to no-slip conditions. The implementation is online and comes with a manual. It is therefore possible to use it for real-life applications such as pipeline or air conditioning, etc.
In the second chapter, I describe a way to model fluid flow through an aquaporine. After making the fluid model precise, the existence of an optimal shape for the dissipated energy criterion is proven. Partial boundary conditions make appear difficulties in the sensitivity analysis of the optimization problem. A specific numerical treatment is presented to overcome this difficulty. Finally, several numerical examples are presented and commented.

Keywords : Navier-Stokes equations, Stokes equations, Shape optimizaton, Augmented Lagrangian, Aquaporins