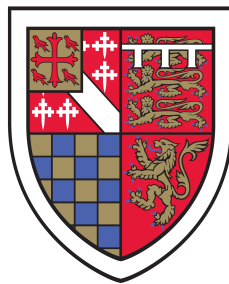


# Stabilization by shape optimization



FIRST YEAR REPORT

C. Hennekinne  
St Edmund's College  
University of Cambridge

**Supervisor**  
Dr. M. P. Juniper

**Advisor**  
Dr. C. P. Caulfield

August 2013



# Contents

<b>Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>iii</b>
<b>Nomenclature</b>	<b>vi</b>
<b>1 Preamble</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Mathematical framework . . . . .	2
1.3 Application . . . . .	4
<b>2 Standard adjoint algorithm</b>	<b>9</b>
2.1 Parameter optimization . . . . .	9
2.2 Shape optimization . . . . .	12
<b>3 Double-decker adjoint algorithm</b>	<b>17</b>
3.1 Modal sensitivity analysis . . . . .	18
3.2 Parameter optimization . . . . .	20
3.3 Shape optimization . . . . .	23
<b>4 Future Work</b>	<b>29</b>
<b>A Mathematical tools</b>	<b>33</b>
A.1 Boundary residual . . . . .	33
A.2 Green's identities . . . . .	35
<b>B On the Navier–Stokes equations</b>	<b>37</b>
B.1 Geometry and notation . . . . .	37
B.2 Boundary condition . . . . .	37

**CONTENTS**

---

B.3 Conservative form . . . . . 38  
B.4 Advective form . . . . . 39  
**Bibliography** . . . . . **41**

# List of Figures

1.1	Geometry of the backward facing slope . . . . .	4
1.2	Baseflow with $Re = 700$ . . . . .	5
1.3	Leading mode with $Re = 700$ and $\beta = 1.0$ . . . . .	5
1.4	Shape of the slope for $s \in \{0.5, 1, 1.5\}$ . . . . .	7
2.1	Diagram of the standard adjoint algorithm . . . . .	11
2.2	Validation of the $\alpha$ -parameter gradient of the cost function . . . . .	13
2.3	Validation of the shape gradient of the cost function . . . . .	16
3.1	Diagram of the modal sensitivity analysis . . . . .	19
3.2	Validation of the $\beta$ -parameter gradient of the growth rate . . . . .	21
3.3	Diagram of the double-decker adjoint algorithm . . . . .	22
3.4	Validation of the $\alpha$ -parameter gradient of the growth rate . . . . .	23
3.5	Validation of the shape gradient of the growth rate . . . . .	27



# Nomenclature

## Common symbols

Symbol	Description	Definition
$\underline{\underline{D}}$	Definition	
$\mathbf{1}_{\mathcal{E}}$	Characteristic function of $\mathcal{E}$	
$f _{\mathcal{E}}$	Restriction of a function on a subset $\mathcal{E}$	
$\overline{\mathcal{E}}^{\ \cdot\ _{\mathcal{H}}}$	Closure of the set $\mathcal{E}$ in the normed space $\mathcal{H}$	
$\overline{\nabla}$	Symmetric gradient	$\frac{1}{2}(\nabla \cdot + \nabla^t \cdot)$

## Sets

Symbol	Description	Definition / Properties
$\mathbb{R}$	Field of real numbers	
$\Omega$	Bounded open domain of dimension $n$	$\subseteq \mathbb{R}^n$
$\Gamma$	Boundary	$= \partial\Omega, = \Gamma^d \cup \Gamma^\sigma, = \Gamma_+ \cup \Gamma_-$
$\Gamma^d$	Dirichlet boundary : part of the boundary where a value is prescribed	$\subseteq \Gamma, \Gamma^d \cap \Gamma^\sigma = \emptyset$
$\Gamma^\sigma$	Flux boundary : part of the boundary where a flux is prescribed	$\subseteq \Gamma, \Gamma^d \cap \Gamma^\sigma = \emptyset$
$\overline{\Gamma}^d$	Moving boundary : part of the Dirichlet boundary that is allowed to deform when performing shape modification	$\subseteq \Gamma^d$

## Nomenclature

---

$\Gamma_+$	Inflow boundary for a vector field $\mathbf{u}$	$= \{\mathbf{x} \in \Gamma \mid \mathbf{u} \cdot \mathbf{n} < 0\}, \subseteq \Gamma$
$\Gamma_-$	Outflow boundary for a vector field $\mathbf{u}$	$= \{\mathbf{x} \in \Gamma \mid \mathbf{u} \cdot \mathbf{n} \geq 0\}, \subseteq \Gamma$

### Functions spaces

Symbol	Description	Definition
$H^k(\Omega)$	Sobolev space of degree $k$ equipped with the 2-norm	
$H_0^k(\Omega)$	Sobolev space with zero Dirichlet boundary condition	$= \{f \in H^k(\Omega) \mid f _{\Gamma} = 0\}$
$\bar{\mathcal{S}}$	Vectorial Sobolev space of degree 1	$= H^1(\Omega)^{\bar{k}}$
$\bar{\mathcal{S}}_0$	Vectorial Sobolev space of degree 1 with zero Dirichlet boundary condition	$= \{\mathbf{s} \in \bar{\mathcal{S}} \mid \mathbf{s} _{\Gamma^d} = \mathbf{0}\}$
$\bar{\mathcal{S}}[\mathbf{g}]$	Vectorial Sobolev space of order 1 with specified Dirichlet boundary condition	$= \{\mathbf{s} \in \bar{\mathcal{S}} \mid \mathbf{s} _{\Gamma^d} = \mathbf{g}\}$
$\dot{\mathcal{S}}$	Vectorial Sobolev space of order 0	$= L^2(\Omega)^{\dot{k}}$
$\mathcal{S}$	Mixed element space	$= \bar{\mathcal{S}} \times \dot{\mathcal{S}}$
$\mathcal{S}_*$	Mixed element space with Dirichlet boundary condition	$= \bar{\mathcal{S}}_* \times \dot{\mathcal{S}}$
$\mathcal{S}_*^2$	Mixed element space of higher regularity	$= (\bar{\mathcal{S}}_* \cap H^2(\Omega)^{\bar{k}}) \times (\dot{\mathcal{S}} \cap H^1(\Omega)^{\dot{k}})$

# Chapter 1

## Preamble

### 1.1 Motivation

A component of the engineering design process consists of shaping forms so that they have desired properties. Recent years have seen hopes of an automation of part of this process. For system of limited complexity, the design can be formulated as the minimization of a cost function, such as the viscous dissipation, by changing the shape of the domain in which we solve the equations. As the problem may not be convex, there is little hope to find the optimal shape, the shape that reaches the global minimum. However optimization approach such as gradient descent or quasi-Newton method can be applied to improve an initial configuration until reaching a local minimum. This is precisely the goal of shape optimization algorithms. Most of these algorithms rely on adjoint methods to compute the gradient of the objective function with respect to a change of the shape, called the shape derivative. Chapter 2 is dedicated to describing an adjoint-based shape optimization algorithm that uses a parameter-free<sup>1</sup> representation of the boundary. This algorithm will be called the standard adjoint algorithm in this report.

Shape optimization algorithms were developed by the academic community almost two decades ago and are primarily used in structural engineering and fluid mechanics. Attempts are now made to applied these techniques to larger scale systems. In fluid mechanics, the main deterrents to industrial applications are issues related to turbulence. Some of them are briefly discussed in Chapter 4.

Among the limitations of this standard adjoint algorithm is the form of the cost

---

<sup>1</sup>Despite the name, a parametrization of the shape is still introduce when meshing the boundary.

## 1. Preamble

---

function. Indeed, cost functions must be expressed as an integral quantity, i.e. a functional of the solution of the main equation<sup>1</sup>. Although most of the desired properties, such as the drag or the drop in pressure, can be represented in this form, some quantities of interest can only be determined by solving an additional equation. Among them is the stability of the system, which can be quantified by the growth rate of the leading mode<sup>2</sup>. This growth rate is solution of an eigenproblem formed from the solution of the main equation but cannot be expressed as a functional of this solution. Yet, the stability of the system can be of primary importance. From a computational perspective, the instability of the physical system is not always captured by a simulation of a flow because of some incorrect assumptions made on this flow. For example, a steady simulation of a cylinder flow in which we assume a symmetry with respect to the center line is no longer valid passed the first instability as this instability breaks both the steadiness and the symmetry. On a more physical point of view, it is sometimes desirable that the system be stable. For example, we may want to avoid the formation of Karman-vortex street in the wake of an object because it is known to be a cause of vibration in the structure.

Hence, a desirable improvement of the standard adjoint algorithm would enable the inclusion of the growth rate of an auxiliary eigenproblem either as an element of the cost function or as a constraint. This extension that we named double-decker adjoint algorithm is presented in Chapter 3

The principal purpose of this report is to describe the algorithms and establish the various formulae needed to compute the gradients. This is done in Chapter 2 and 3 with an important technical result in Appendix A. Chapter 4 gathers some ideas for future work, both immediate prolongation of the current work or more challenging path.

## 1.2 Mathematical framework

Although the main motivation comes from fluid systems, this method aims to be more generally applied to shape optimization of non-linear systems in which stability is either involved in the cost function or as constraint. For this purpose, derivations will remain quite general and will only be applied to an incompressible fluid system as a next step.

---

<sup>1</sup>Also called direct equation

<sup>2</sup>The system is stable if the growth rate is strictly negative and unstable else.

---

While most of the derivation related to adjoint methods are established with the strong form of the equations, we decided to work only with the weak formulations. This unconventional<sup>1</sup> choice is motivated in several ways:

- The Finite Element Method solves the weak formulation of an equation. The weak solution is evaluated by approximating the evaluation space by a subspace of finite dimension.
- Forming the adjoint problems from the weak formulation does not require the introduction of an adjoint operator in the mathematical sense<sup>2</sup>. When working with the weak formulation, adjoint equations are obtained by interchanging the arguments of the bilinear form which defines the weak formulation (cf. Chapter 2). Thus the same level of regularity is imposed on the direct and adjoint solution, which is not the case with strong forms<sup>3</sup>.
- The handling of boundary conditions is simpler using the weak forms : Dirichlet boundary condition are included in the function space and Neumann or Robin boundary condition are assimilated in the bilinear form. Hence we do not need to derive the appropriate boundary conditions for the strong form of the adjoint equation, which do not necessarily have a clear physical meaning.
- Adjoint methods are used primarily as tools to obtain various gradients. For instance, we do not look at the decomposition of the non-normality in component-type and convective-type [cf. 30, page 7] so the strong form of the adjoint is of little interest.
- The existence of a solution to the three dimensional incompressible Navier–Stokes equations is only guaranteed when working with the standard weak form [19]. This motive is futile as existence problems rarely affect computational simulations of fluid flows.

A shallow look at the derivation written in this report may lead to believe that strong form are never used. This is quite misleading as the boundary residual formulae (A.7) necessary to derive the shape derivative (2.14) and (3.16) are closely linked to

---

<sup>1</sup>Unconventional in the fluid mechanics community, not so much in the applied mathematics one.

<sup>2</sup>Given two Banach space  $B_1$ ,  $B_2$  and their dual space  $B_1^*$ ,  $B_2^*$ , the adjoint (also called dual) of an operator  $T : B_1 \rightarrow B_2$  is the mapping between the dual spaces  $T^\dagger : B_2^* \rightarrow B_1^*$  satisfying  $\forall(x, y^*) \in (B_1, B_2^*) : y^*(T(x)) = T^\dagger(y^*)(x)$

<sup>3</sup>For example, if  $T : H^2(\Omega) \rightarrow L^2(\Omega)$ , then  $T^\dagger : L^2(\Omega) \rightarrow H^{-2}(\Omega)$  and an adjoint equation has to be interpreted as an equality in  $H^{-2}(\Omega)$ . This is obviously a problem for computation given the lack of regularity of  $H^{-2}(\Omega)$ .

## 1. Preamble

---

the strong form of the adjoint equation.

Functions spaces play an important role with the weak formulation. In this report, the general derivation will take place in a mixed function space  $\mathcal{S} = \bar{\mathcal{S}} \times \dot{\mathcal{S}}$ . The space  $\bar{\mathcal{S}}$  is associated to the portion of the state variable on which a Dirichlet condition is imposed on the boundary  $\Gamma^d$  while the space  $\dot{\mathcal{S}}$  is associated to the rest of the state variable on which no Dirichlet condition is imposed. For example, with the incompressible Navier–Stokes equations, the former is the function space of the velocity and the latter is the function space of the pressure.

Shape optimization techniques requires the existence of additional boundary quantities. For example  $u_p|_{\Gamma^d}$ , restriction of the pressure on the boundary, or  $\partial_n \mathbf{u}|_{\Gamma^d}$ , restriction of the normal derivative of the velocity on the boundary, must exist when performing shape optimization using the incompressible Navier–Stokes equations. This quantities do not exist in the function space  $\mathcal{S}$  in which the weak form is defined. For that reason, we will have to assume that some solutions of equations belong to a function space  $\mathcal{S}^2$  with one additional degree of regularity.

Before deriving the formulae necessary for shape optimization, we will study a simpler problem called parameter optimization. Instead of optimizing with respect to the shape of the domain, it consists in optimizing with respect to a parameter that appears in the equation such as the Reynolds number in the Navier–Stokes equations. The formulae derived have little interest but it is a way to introduce adjoint methods using the weak formulation without the added complexity caused by shape optimization.

### 1.3 Application

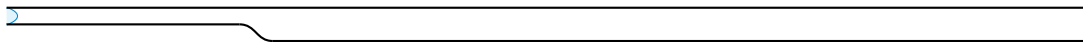


Figure 1.1: Geometry of the backward facing slope

The formulae obtained will be applied to the flow over a backward facing slope with the geometry depicted in Figure 1.1. A Poiseuille flow is imposed at the inlet and an advective boundary condition (B.2) at the outlet. For the range of Reynolds number considered, the steady flow presents two main recirculation bubbles: one behind the slope and another one attached on the upper wall. One of this baseflow is displayed in

Figure 1.2.

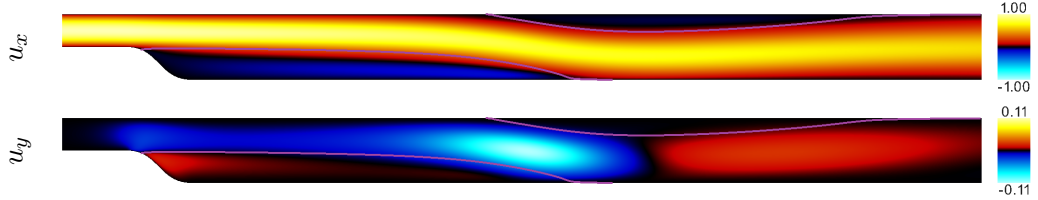


Figure 1.2: Baseflow for a Reynolds number of 700. The velocity  $\mathbf{u} = u_x \mathbf{e}_x + u_y \mathbf{e}_y$  is displayed. The purple line delimits the recirculation bubbles.

For a Reynolds number of about 700, the flow becomes unstable to a three dimensional steady bifurcation with a spanwise wave number of approximately 0.9. The shape of the leading mode is depicted in Figure 1.3.

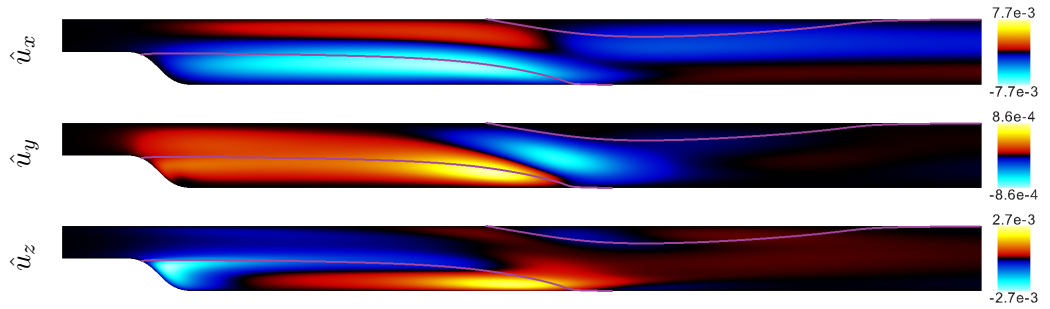


Figure 1.3: Leading mode for a Reynolds number of 700 and a spanwise wave number of 1.0. The velocity  $\hat{\mathbf{u}} = \hat{u}_x \cos(\beta z) \mathbf{e}_x + \hat{u}_y \cos(\beta z) \mathbf{e}_y + \hat{u}_z \sin(\beta z) \mathbf{e}_z$  of this unstable mode is displayed. The purple line delimits the recirculation bubbles of the baseflow.

In order to validate the formulae involving shape modification, we introduce changes in the slope. Let arbitrarily introduce a set of slopes parametrized by a scalar variable  $s$  and having the following parametric representation<sup>1</sup>:

$$\mathcal{B}(s) : [0, 1] \rightarrow \mathbb{R}^2$$

$$t \mapsto \begin{pmatrix} (2t - 1)((3s - 2)t^2 + (-3s + 2)t + 1) \\ t^2(2t - 3) \end{pmatrix} \quad (1.1)$$

A quantification of the instantaneous displacement of the slope as  $s$  varies is needed to

<sup>1</sup>This is the parametric representation of a uniform B-spline [4] with knots  $(0, 0, 0, 0, 1, 1, 1, 1)$  and control points  $((-1, 0)^t, (-1 + s, 0)^t, (1 - s, -1)^t, (1, -1)^t)$ .

## 1. Preamble

---

compute shape derivatives. Let us introduce the flow mapping:

$$\begin{aligned} \Phi(s) : \text{Im } \mathcal{B}(s) \times \mathbb{R} &\rightarrow \mathbb{R}^2 \\ ((x, y)^t, u) &\mapsto \mathcal{B}(s + u) (\mathcal{B}(s)^{-1}((x, y)^t)) \end{aligned} \quad (1.2)$$

The only purpose of this application is to define the boundary displacement as  $\mathbf{h}_{\mathcal{B}(s)} = \partial_u \Phi(s)|_{u=0}$ . Using the parametric representation of  $\mathcal{B}(s)$ , we can obtain a parametric representation of  $\mathbf{h}_{\mathcal{B}(s)}$ :

$$\begin{aligned} \mathbf{h}_{\mathcal{B}(s)} : \text{Im } \mathcal{B}(s) &\rightarrow \mathbb{R}^2 \\ (x, y)^t &\mapsto \begin{pmatrix} 3t(t-1)(2t-1) \\ 0 \end{pmatrix} \text{ with } t = \mathcal{B}(s)^{-1}((x, y)^t) \end{aligned} \quad (1.3)$$

The boundary displacement is dependent of the parametrization of the curve and is therefore not a physical quantity. As we will see in this report the boundary displacement only appears in shape derivatives problem through the normal boundary displacement  $\mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)}$ , which does not depend on the parametrization of  $\mathcal{B}(s)$ . The exterior normal to  $\mathcal{B}(s)$  can be computed using the definition (1.1):

$$\begin{aligned} \mathbf{n}_{\mathcal{B}(s)} : \text{Im } \mathcal{B}(s) &\rightarrow \mathbb{R}^2 \\ (x, y)^t &\mapsto \frac{\begin{pmatrix} 6t(t-1) \\ -6(3s-2)t^2 + 6(3s-2)t - 3s \end{pmatrix}}{\sqrt{(6(3s-2)t^2 - 6(3s-2)t + 3s)^2 + (6t(t-1))^2}} \\ &\text{with } t = \mathcal{B}(s)^{-1}((x, y)^t) \end{aligned} \quad (1.4)$$

Hence a parametric representation of the normal boundary displacement is:

$$\begin{aligned} \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} : \text{Im } \mathcal{B}(s) &\rightarrow \mathbb{R} \\ (x, y)^t &\mapsto \frac{18t^2(t-1)^2(2t-1)}{\sqrt{(6(3s-2)t^2 - 6(3s-2)t + 3s)^2 + (6t(t-1))^2}} \\ &\text{with } t = \mathcal{B}(s)^{-1}((x, y)^t). \end{aligned} \quad (1.5)$$

This normal boundary displacement will appear in Section 2.2 and 3.3.

This setup has now three parameters, a shape parameter  $s$ , the Reynolds number  $\text{Re}$  and the spanwise wave number  $\beta$  when dealing with stability. The reference configuration is chosen to be the one with  $s = 1$ ,  $\text{Re} = 700$  and  $\beta = 1$ .

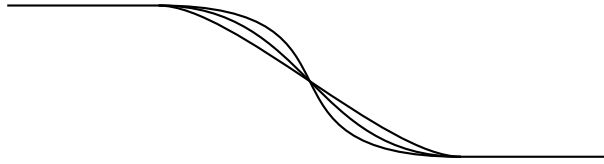


Figure 1.4: Shape of the slope for  $s \in \{0.5, 1, 1.5\}$

To validate the various formulae derived in this report, we make one of the three parameters vary around the reference configuration and then compare the value of the gradient obtained using adjoint methods with the one obtained by finite differences.

In order to roughly estimate the influence of the resolution requirement, two different meshes were built for each value of  $s$  investigated: a coarse mesh with approximately 10 000 cells<sup>1</sup> and a fine mesh with around 100 000 cells. Both the coarse and fine mesh have a higher density of cells around the slope. Mixed finite elements were used with continuous Lagrange elements of order 2 for the velocity and continuous Lagrange elements of order 1 for the pressure<sup>2</sup>.

---

<sup>1</sup>Triangles that compose the mesh.

<sup>2</sup>This finite elements are often called Taylor–Hood finite element.



## Chapter 2

# Standard adjoint algorithm

Pironneau [24, 25] was the first to investigate shape optimality problems in fluid mechanics using adjoint methods. He derived an expression for the shape derivative of the viscous dissipation and deduced the necessary condition for the optimality of a shape. However, he could only apply it to elementary problems in which the optimal shape could be analytically described. Years later, Jameson gave a second life to Pironneau's idea by designing a computer algorithm to get a locally optimal shape. Along the years, he applied this algorithm to fluid flows of increasing complexity from potential flow [14] to flow governed by the incompressible Navier–Stokes equations [15].

Recent years have seen improvements in two separate directions. On the one hand, some researchers have applied these techniques to industrial scale systems. For example, Schmidt *et al.* [29] optimized the shape of a three dimensional airfoil under transonic conditions using Euler's equation. This approach requires highly optimized code and large computational power. On the other hand, other researchers focused on systems of reasonable dimension and performed an optimization with a less conventional cost function such as the intensity of the far field sonic boom [23] or the noise emitted by a flow [22]. The development of high level finite element code makes it now possible to perform shape optimization on a variety of different equations and cost functions, from microfluidic systems to fluids with chemistry.

## 2.1 Parameter optimization

### 2.1.1 Derivation

Consider a functional  $\mathcal{N}_\alpha : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ , linear in its second argument and parametrized by  $\alpha \in \mathcal{A}$ . Consider also the boundary function  $\mathbf{g} \in H^1(\Gamma^d)^k$  and define the direct

## 2. Standard adjoint algorithm

---

state  $\mathbf{x}$  as the solution of the direct equation:

$$\mathbf{x} \in \mathcal{V}^2[\mathbf{g}] \text{ s.t. } \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{N}_\alpha(\mathbf{x}, \mathbf{v}) = 0. \quad (2.1)$$

Let also define the set of direct states  $\mathcal{X} = \{\mathbf{x} \in \mathcal{V}[\mathbf{g}] \mid \exists \alpha \in \mathcal{A} : \mathbf{x} \text{ solution of (2.1)}\}$ .

Consider the functional  $\mathcal{J} : \mathcal{V} \rightarrow \mathbb{R}$  called cost function, independent of  $\alpha$  and composed of a volume<sup>1</sup> integral:

$$\forall \mathbf{v} \in \mathcal{V} : \mathcal{J}(\mathbf{x}) = \int_{\Omega} j(\mathbf{v}) \quad (2.2)$$

Our goal is to find the minimum of the set  $\mathcal{J}(\mathcal{X}) = \{\mathcal{J}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$  over the set of parameters  $\alpha \in \mathcal{A}$ . As the problem may not be convex, we satisfy ourselves with finding local optima using an optimization algorithm. Starting with an initial parameter  $\alpha$  and a state  $\mathbf{x} \in \mathcal{X}$ , we wish to evaluate the gradient  $d_\alpha \mathcal{J}(\mathbf{x})$ .

Differentiating the definition (2.2) with respect to  $\alpha$  leads to an expression of this gradient in terms of the derivative of the direct state  $d_\alpha \mathbf{x}$ :

$$d_\alpha \mathcal{J}(\mathbf{x}) = \int_{\Omega} \nabla j[\mathbf{x}](d_\alpha \mathbf{x}) \stackrel{\mathcal{D}}{=} \nabla \mathcal{J}[\mathbf{x}](d_\alpha \mathbf{x}). \quad (2.3)$$

As soon as  $\alpha$  is composed of a couple of variables, adjoint methods become less expensive than finite differences to compute the gradient. To apply an adjoint method, we need first to know the equation verified by the derivative of the direct state  $d_\alpha \mathbf{x}$ .

Let us introduce the bilinear<sup>2</sup> form  $\mathcal{L}_\alpha[\mathbf{x}] : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ , called the linearized functional and defined by<sup>3</sup>  $\forall \mathbf{v} \in \mathcal{S} : \mathcal{L}_\alpha[\mathbf{v}] = \partial_1 \mathcal{N}_\alpha(\mathbf{v})$ . Differentiating the direct equation (2.1) with respect to  $\alpha$  leads to the desired equation. The derivative of the direct state  $d_\alpha \mathbf{x}$  satisfies the equation:

$$d_\alpha \mathbf{x} \in \mathcal{V}_0 \text{ s.t. } \forall \mathbf{v} \in \mathcal{V}_0 : \partial_\alpha \mathcal{N}_\alpha(\mathbf{x}, \mathbf{v}) + \mathcal{L}_\alpha[\mathbf{x}](d_\alpha \mathbf{x}, \mathbf{v}) = 0 \quad (2.4)$$

Adjoint method consists of solving an additional carefully chosen linear equation. De-

---

<sup>1</sup>A cost function composed of a volume and a surface integral will be investigated soon.

<sup>2</sup>Linear in each of its two arguments.

<sup>3</sup> $\partial_1$  denotes the partial derivative with respect to the first argument.

find the adjoint state  $\mathbf{y}$  as the solution of the adjoint equation<sup>1</sup>:

$$\mathbf{y} \in \mathcal{V}_0^2 \text{ s.t. } \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_\alpha[\mathbf{x}](\mathbf{v}, \mathbf{y}) = \nabla \mathcal{J}[\mathbf{x}](\mathbf{v}). \quad (2.5)$$

The adjoint equation (2.5) is verified for every test function  $\mathbf{v}$  in  $\mathcal{V}_0$ . As  $d_\alpha \mathbf{x} \in \mathcal{V}_0$ , we have in particular the following relation:

$$d_\alpha \mathcal{J}(\mathbf{x}) = \nabla \mathcal{J}[\mathbf{x}](d_\alpha \mathbf{x}) = \mathcal{L}_\alpha[\mathbf{x}](d_\alpha \mathbf{x}, \mathbf{y}). \quad (2.6)$$

Similarly, the equation (2.4) is verified for every test function  $\mathbf{v}$  in  $\mathcal{V}_0$ . As  $\mathbf{y} \in \mathcal{V}_0$ , we end up with the following expression for the gradient  $d_\alpha \mathcal{J}(\mathbf{x})$ :

<b><math>\alpha</math>-parameter gradient of the cost function</b>
$d_\alpha \mathcal{J}(\mathbf{x}) = -\partial_\alpha \mathcal{N}_\alpha(\mathbf{x}, \mathbf{y}) \quad (2.7)$

Assuming that the analytical expression of the functionals  $\mathcal{N}_\alpha$ ,  $\mathcal{L}_\alpha[\mathbf{x}]$ ,  $\nabla \mathcal{J}[\mathbf{x}]$  and  $\partial_\alpha \mathcal{N}_\alpha$  are known, the gradient  $d_\alpha \mathcal{J}(\mathbf{x})$  can be evaluated by solving the direct equation (2.1), solving the adjoint equation (2.5) and forming the expression (2.7). The gradient can then be used to update the parameter in an optimization algorithm. This process is schematized in Figure 2.1.

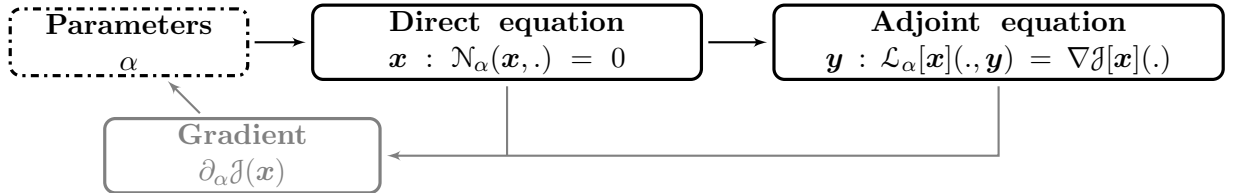


Figure 2.1: Diagram of the standard adjoint algorithm

### 2.1.2 Application

The problem described in Section 1.3 is a two-dimensional incompressible flow problem. An appropriate function space for this problem is  $\mathcal{V} = H^1(\Omega)^2 \times L^2(\Omega)$  with a state variable composed of velocity and pressure. The direct equation is the weak form of the incompressible Navier–Stokes equation. As an advective boundary condition is imposed at the outlet, it is simpler to work with the advection form of the equation

<sup>1</sup>Note that the test function  $\mathbf{v}$  is the first argument of  $\mathcal{L}_\alpha[\mathbf{x}]$  which is how adjoint equations are obtained in the weak formulation framework.

## 2. Standard adjoint algorithm

---

(cf. Appendix B). The non-linear functional  $\mathcal{N}_\alpha$  takes the form:

$$\mathcal{N}_\alpha : ((\mathbf{u}, u_p), (\boldsymbol{\eta}, \eta_p)) \mapsto \int_{\Omega} \nabla \mathbf{u} : \left( \boldsymbol{\eta} \otimes \mathbf{u} + \frac{\nabla \boldsymbol{\eta}}{\alpha} \right) - u_p \operatorname{div} \boldsymbol{\eta} + \eta_p \operatorname{div} \mathbf{u} \quad (2.8)$$

where  $\alpha$  is the Reynolds number.

Taking the derivative of this functional with respect to  $\alpha$  is straightforward:

$$\partial_\alpha \mathcal{N}_\alpha : ((\mathbf{u}, u_p), (\boldsymbol{\eta}, \eta_p)) \mapsto -\frac{1}{\alpha^2} \int_{\Omega} \nabla \mathbf{u} : \nabla \boldsymbol{\eta} \quad (2.9)$$

Differentiating with respect to  $(\mathbf{u}, u_p)$  is also accessible and leads to an expression for the linearized functional:

$$\begin{aligned} \mathcal{L}_\alpha [(\mathbf{u}, u_p)] : ((\mathbf{v}, v_p), (\boldsymbol{\eta}, \eta_p)) \mapsto & \int_{\Omega} \nabla \mathbf{v} : \left( \boldsymbol{\eta} \otimes \mathbf{u} + \frac{\nabla \boldsymbol{\eta}}{\alpha} \right) + \nabla \mathbf{u} : \boldsymbol{\eta} \otimes \mathbf{v} \\ & - v_p \operatorname{div} \boldsymbol{\eta} + \eta_p \operatorname{div} \mathbf{v} \end{aligned} \quad (2.10)$$

The cost function we are interested in is the viscous dissipation which has the following expression:

$$\mathcal{J} : ((\mathbf{u}, u_p)) \mapsto \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{u} \quad (2.11)$$

Finally, the gradient of this cost function is:

$$\nabla \mathcal{J}[(\mathbf{u}, u_p)] : ((\boldsymbol{\eta}, \eta_p)) \mapsto 2 \int_{\Omega} \nabla \mathbf{u} : \nabla \boldsymbol{\eta} \quad (2.12)$$

After all these definitions and analytical derivations, it is possible to perform one loop of the standard adjoint algorithm in order to compute  $d_\alpha \mathcal{J}((\mathbf{u}, u_p))$ . As seen in Figure 2.2, the gradient obtained using the adjoint method is in good agreement with the gradient approximated by finite differences.

## 2.2 Shape optimization

If a field is a solution of an equation on a domain, changing the shape of the domain will affect the field. The shape derivative  $d_{\mathbf{h}}$  of a field corresponds to the infinitesimal change of that field when the boundary of the domain is infinitesimally moved in the direction of  $\mathbf{h}$ . There are two ways to introduce this concept mathematically. The interior variation approach introduces a deformation of the whole domain that matches

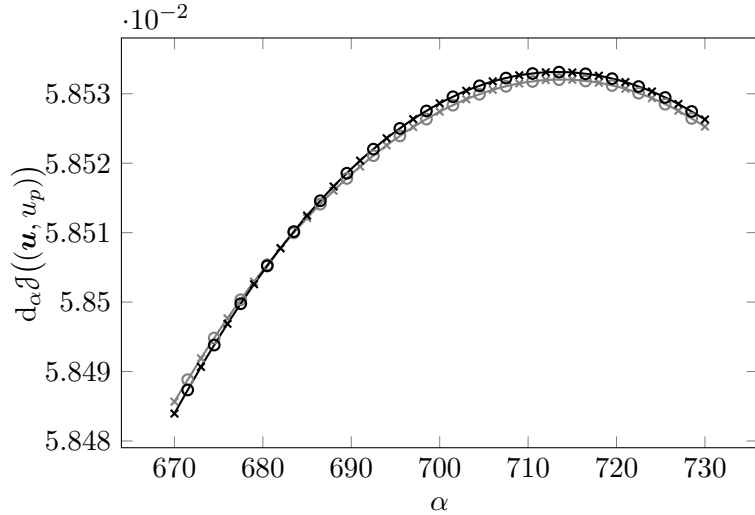


Figure 2.2: Validation of the formula of the  $\alpha$ -parameter gradient of the cost function (2.7) using both a coarse mesh (gray) and a fine mesh (black). The circles corresponds to the gradient obtained with finite differences while the crosses and the fitting spline represents the gradient computed using the adjoint method.

the boundary displacement  $\mathbf{h}$  on the boundary<sup>1</sup>. The moving surface approach does not introduce a deformation of the domain but relies on advanced mathematical results on moving manifolds.

All these mathematical intricacies will be avoided in this report and we will only use some technical results on how to compute the shape derivative [12]. More specifically, we will need to know how to find the equation that is satisfied by the shape derivative of a field knowing the equation satisfied by the field in question (cf. equation (2.14)) and how to compute the shape derivative of a functional that depends on a field (cf. expression (2.13)).

### 2.2.1 Derivation

Consider again the functional  $\mathcal{N}_\alpha : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  and the direct state  $\mathbf{x}$  solution of the direct equation (2.1). Consider also the cost function  $\mathcal{J}$  defined in (2.2). The shape derivative of this cost function is:

$$d_{\mathbf{h}}\mathcal{J}(\mathbf{x}) = \int_{\Omega} \nabla j[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}) + \int_{\Gamma^d} j(\mathbf{x})(\mathbf{h} \cdot \mathbf{n}) \stackrel{\mathcal{D}}{=} \nabla \mathcal{J}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}) + \int_{\Gamma^d} j(\mathbf{x})(\mathbf{h} \cdot \mathbf{n}) \quad (2.13)$$

Since the boundary displacement  $\mathbf{h}$  belongs to a space of infinite dimension, it is

<sup>1</sup>Such deformation is not unique so it raises questions on the existence of this derivative

## 2. Standard adjoint algorithm

---

not even possible to evaluate the shape derivative of the direct state  $d_{\mathbf{h}}\mathbf{x}$  by finite differences without introducing a parameterization of the shape. Adjoint methods are well adapted in this case but require the knowledge of the equation satisfied by  $d_{\mathbf{h}}\mathbf{x}$ . This equation can be obtained by using some shape derivative techniques on the direct equation (2.1). The shape derivative of the direct state  $d_{\mathbf{h}}\mathbf{x}$  is solution of the equation:

$$d_{\mathbf{h}}\mathbf{x} \in \mathcal{V}^2 \left[ -\mathbf{1}_{\overline{\Gamma}^d}((\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}} \bar{\mathbf{x}}|_{\overline{\Gamma}^d}) \right] \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_\alpha[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{v}) = 0 \quad (2.14)$$

Define the adjoint state  $\mathbf{y}$  as the solution of the adjoint equation<sup>1</sup>:

$$\mathbf{y} \in \mathcal{V}_0^2 \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_\alpha[\mathbf{x}](\mathbf{v}, \mathbf{y}) = \nabla \mathcal{J}[\mathbf{x}](\mathbf{v}) \quad (2.15)$$

Unlike the parameter optimization case, the shape derivative of the direct state  $d_{\mathbf{h}}\mathbf{x}$  does not belong to the space  $\mathcal{V}_0$ ;  $d_{\mathbf{h}}\mathbf{x}$  has a non zero Dirichlet boundary condition on  $\overline{\Gamma}^d$ . However, the equation (2.15) can be extended to the space  $\mathcal{V}$  by adding some terms and provided that the operators  $\mathcal{L}_\alpha[\mathbf{x}]$  and  $\nabla \mathcal{J}[\mathbf{x}]$  admits a specific decomposition. The mathematical details have been pushed in Appendix A for clarity. Using the boundary residual formula (A.7), we obtain the expression of the shape derivative:

### Shape gradient of the cost function

$$d_{\mathbf{h}}\mathcal{J}(\mathbf{x}) = \langle j(\mathbf{x})|_{\overline{\Gamma}^d}, \mathbf{h} \cdot \mathbf{n} \rangle_{L^2(\overline{\Gamma}^d)} + \langle \gamma^\dagger \{ \mathcal{L}_\alpha[\mathbf{x}] \}(\mathbf{y}) - \gamma \{ \nabla \mathcal{J}[\mathbf{x}] \}, (\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}} \bar{\mathbf{x}}|_{\overline{\Gamma}^d} \rangle_{L^2(\overline{\Gamma}^d)} \quad (2.16)$$

If the analytical expression of the functionals  $\mathcal{L}_\alpha[\mathbf{x}]$  and  $\nabla \mathcal{J}[\mathbf{x}]$  are known, Green's identities should be used to decompose the functionals in the appropriate form (A.3). These decompositions will define  $\gamma^\dagger \{ \mathcal{L}_\alpha[\mathbf{x}] \}$  and  $\gamma \{ \mathcal{J}[\mathbf{x}] \}$ . The gradient  $d_{\mathbf{h}}\mathcal{J}(\mathbf{x})$  can then be evaluated by solving the direct equation (2.1), solving the adjoint equation (2.15) and forming the expression (2.16).

### 2.2.2 Application

Going back to the problem introduced in Section 1.3, the expression of  $\gamma \{ \nabla \mathcal{J}[(\mathbf{u}, u_p)] \}$  and  $\gamma^\dagger \{ \mathcal{L}_\alpha[(\mathbf{u}, u_p)] \}$  can be derived from the expression (2.10) and (2.12) using Green's

---

<sup>1</sup>Note that this adjoint equation is identical to the one introduced for the parameter optimization. The adjoint equation would have been different if the cost function was also containing a surface integral.

---

identities.

Using the Green's identity (A.10b) leads to:

$$\nabla \mathcal{J}[(\mathbf{u}, u_p)]((\boldsymbol{\eta}, \eta_p)) = -2 \int_{\Omega} \Delta \mathbf{u} \cdot \boldsymbol{\eta} + 2 \int_{\Gamma} \partial_{\mathbf{n}} \mathbf{u} \cdot \boldsymbol{\eta} \quad (2.17)$$

This form matches the decomposition (A.3a) so by definition:

$$\gamma \{ \nabla \mathcal{J}[(\mathbf{u}, u_p)] \} = \partial_{\mathbf{n}} \mathbf{u} \quad (2.18)$$

Similarly, using the Green's identities (A.9), (A.10a) and (A.10b):

$$\begin{aligned} \mathcal{L}_{\alpha} [(\mathbf{u}, u_p)]((\mathbf{w}, w_p), (\boldsymbol{\eta}, \eta_p)) &= \int_{\Omega} - \left( \operatorname{div}(\boldsymbol{\eta} \otimes \mathbf{u}) + \frac{\Delta \boldsymbol{\eta}}{\alpha} \right) \cdot \mathbf{w} + \nabla \mathbf{u} : \boldsymbol{\eta} \otimes \mathbf{w} \\ &+ \int_{\Omega} -w_p \operatorname{div} \boldsymbol{\eta} - \nabla \eta_p \cdot \mathbf{w} + \int_{\Gamma} (\mathbf{w} \cdot \boldsymbol{\eta})(\mathbf{u} \cdot \mathbf{n}) + \frac{\nabla \boldsymbol{\eta}}{\alpha} : \mathbf{w} \otimes \mathbf{n} + \eta_p (\mathbf{w} \cdot \mathbf{n}) \end{aligned} \quad (2.19)$$

This form matches the decomposition (A.3b) so by definition:

$$\gamma^{\dagger} \{ \mathcal{L}_{\alpha} [(\mathbf{u}, u_p)] \}((\boldsymbol{\eta}, \eta_p)) = (\mathbf{u} \cdot \mathbf{n}) \boldsymbol{\eta} + \alpha^{-1} \partial_{\mathbf{n}} \boldsymbol{\eta} + \eta_p \mathbf{n} = \alpha^{-1} \partial_{\mathbf{n}} \boldsymbol{\eta} + \eta_p \mathbf{n} \quad (2.20)$$

The last equality is due to the zero Dirichlet boundary condition on  $\mathbf{u}$ .

Noting  $(\bar{\mathbf{u}}, \bar{u}_p)$  the direct state and  $(\tilde{\mathbf{u}}, \tilde{u}_p)$  the adjoint state, it is now possible to have an explicit form for the shape derivative:

$$\mathrm{d}_s \mathcal{J}((\bar{\mathbf{u}}, \bar{u}_p)) = \left\langle -(\partial_{\mathbf{n}} \bar{\mathbf{u}})^2 + \alpha^{-1} \partial_{\mathbf{n}} \bar{\mathbf{u}} \cdot \partial_{\mathbf{n}} \tilde{\mathbf{u}} + \tilde{u}_p \partial_{\mathbf{n}} \bar{\mathbf{u}} \cdot \mathbf{n}, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \right\rangle_{L^2(\bar{\Gamma}^d)} \quad (2.21)$$

As the velocity of the direct state  $\bar{\mathbf{u}}$  obeys a zero Dirichlet boundary condition on  $\bar{\Gamma}^d$  and an incompressibility condition, a further simplification is possible since  $\partial_{\mathbf{n}} \bar{\mathbf{u}} \cdot \mathbf{n} = \nabla \bar{\mathbf{u}} : \mathbf{n} \otimes \mathbf{n} = 0^1$ :

$$\mathrm{d}_s \mathcal{J}((\bar{\mathbf{u}}, \bar{u}_p)) = \left\langle -(\partial_{\mathbf{n}} \bar{\mathbf{u}})^2 + \alpha^{-1} \partial_{\mathbf{n}} \bar{\mathbf{u}} \cdot \partial_{\mathbf{n}} \tilde{\mathbf{u}}, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \right\rangle_{L^2(\bar{\Gamma}^d)} \quad (2.22)$$

If we take into account the different definitions of the cost function and adjoint equation, this expression matches the one of Schmidt & Schulz [28].

---

<sup>1</sup>Because of the Dirichlet boundary condition on  $\bar{\mathbf{u}}$ , the derivatives of the velocity along the surface are zero so  $\nabla \bar{\mathbf{u}} = (\nabla \bar{\mathbf{u}} \cdot \mathbf{n}) \otimes \mathbf{n}$ . In particular,  $\nabla \bar{\mathbf{u}} : \mathbf{J} = \nabla \bar{\mathbf{u}} : \mathbf{n} \otimes \mathbf{n}$ . Finally, because of the incompressibility condition  $\nabla \bar{\mathbf{u}} : \mathbf{J} = \operatorname{div} \bar{\mathbf{u}} = 0$ .

## 2. Standard adjoint algorithm

---

For numerical implementation purposes, it may be preferable to use a formulation that does not involve the normal derivative  $\mathbf{n}$ . As the velocities of the direct and adjoint state are null on  $\bar{\Gamma}^d$ , the equalities  $\partial_n \bar{\mathbf{u}} \cdot \partial_n \bar{\mathbf{u}} = \nabla \bar{\mathbf{u}} : \nabla \bar{\mathbf{u}}$  and  $\partial_n \bar{\mathbf{u}} \cdot \partial_n \tilde{\mathbf{u}} = \nabla \bar{\mathbf{u}} : \nabla \tilde{\mathbf{u}}$ <sup>1</sup> valid on the boundary leads to a different version of the formula (2.22):

$$d_s \mathcal{J}((\bar{\mathbf{u}}, \bar{u}_p)) = \langle -\nabla \bar{\mathbf{u}} : \nabla \bar{\mathbf{u}} + \alpha^{-1} \nabla \bar{\mathbf{u}} : \nabla \tilde{\mathbf{u}}, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \rangle_{L^2(\bar{\Gamma}^d)} \quad (2.23)$$

The implementation of this formula results in the values of the gradient  $d_s \mathcal{J}((\bar{\mathbf{u}}, \bar{u}_p))$  displayed in Figure 2.3. For the fine mesh, there is a decent agreement between the gradient obtained with the adjoint method and the finite difference approximation. However, for the coarse mesh, the finite difference approximation of the gradient is completely scattered and the gradient obtained with the adjoint method, albeit more consistency, is way off the converged result.

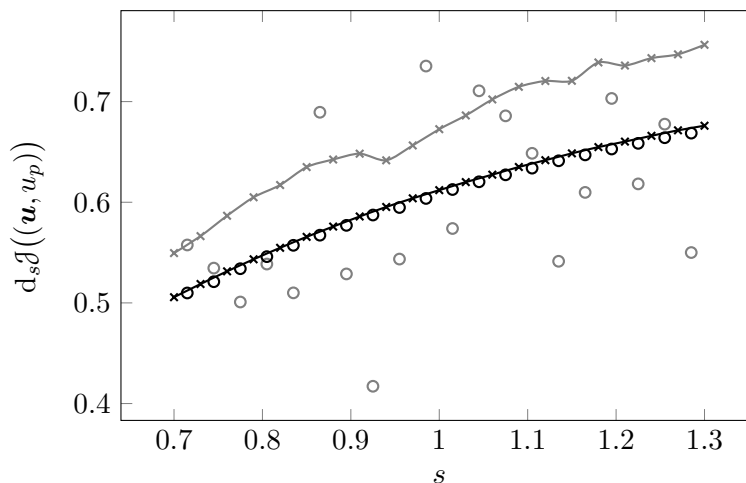


Figure 2.3: Validation of the formula of the shape gradient of the cost function (2.16) using both a coarse mesh (gray) and a fine mesh (black). The circles corresponds to the gradient obtained with finite differences while the crosses and the fitting spline represents the gradient computed using the adjoint method.

---

<sup>1</sup>Because of the Dirichlet boundary condition on  $\bar{\mathbf{u}}$  and  $\tilde{\mathbf{u}}$ ,  $\nabla \bar{\mathbf{u}} = (\nabla \bar{\mathbf{u}} \cdot \mathbf{n}) \otimes \mathbf{n}$  and  $\nabla \tilde{\mathbf{u}} = (\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \otimes \mathbf{n}$  so  $\nabla \bar{\mathbf{u}} : \nabla \tilde{\mathbf{u}} = \partial_n \bar{\mathbf{u}} \cdot \partial_n \tilde{\mathbf{u}}$

## Chapter 3

# Double-decker adjoint algorithm

Optimizing a shape with respect to an eigenvalue is an old mathematical problem that Hadamard [10] pushed forward. Indeed, he derived the shape derivative of eigenvalues of the Laplacian with Dirichlet boundary condition, named afterwards Hadamard's formula<sup>1</sup>. Through the years, the formula was extended to various operators and boundary conditions and the mathematical framework of this problems was refined [3, 9].

While shape derivative are relatively easy to derive and can lead to a locally optimal shape, knowing whether or not this local minimum is a global one is appreciably more complex. For example, the optimal forms of the Laplacian with Dirichlet boundary condition are only known for the two leading eigenvalues [12].

As far as we know, these mathematical techniques have not been applied to fluid flows. These may be due to the fact that most of these shape derivatives were derived for self-adjoint eigenproblem, i.e. the direct and adjoint eigenvectors<sup>2</sup> are identical. But, contrary to the Laplacian, the linearized Navier–Stokes operator is not self-adjoint which means that an adjoint eigenproblem must be introduced, in the manner of modal sensitivity analysis. Moreover, because of the non-linearity of the Navier–Stokes equations, the linearized Navier–Stokes operator has a dependency on the direct state so an additional adjoint equation must also be solved.

Nevertheless the problem of optimizing a shape with a constraint on the stability has already been explored. Heuveline & Strauß [13] studied the minimization of the drag of a cylinder while keeping the flow stable. Because they were computing the shape gradient using finite difference, they had to introduce a rough parametrization of the

---

<sup>1</sup>If  $\varphi \in H_0^2(\Omega)$  is the solution of  $\forall \psi \in H_0^1(\Omega) : \int_{\Omega} \nabla \varphi \cdot \nabla \psi = \sigma \int_{\Omega} \varphi \psi$  then  $d_{\mathbf{h}} \sigma = - \int_{\Gamma} |\nabla \varphi|^2 (\mathbf{h} \cdot \mathbf{n})$

<sup>2</sup>Also called right and left eigenvectors

### 3. Double-decker adjoint algorithm

---

shape and solve at each iteration as many eigenproblems as the number of parameters.

## 3.1 Modal sensitivity analysis

### 3.1.1 Derivation

Consider again the functional  $\mathcal{N}_\alpha : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  parametrized by  $\alpha \in \mathcal{A}$  that is used to define the direct state  $\mathbf{x}$ , solution of the direct equation:

$$\mathbf{x} \in \mathcal{V}^2[\mathbf{g}] \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{N}_\alpha(\mathbf{x}, \mathbf{v}) = 0 \quad (3.1)$$

Consider the bilinear forms  $\Gamma_{\alpha,\beta}[\mathbf{x}] : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$ , parametrized by  $(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}$  and having a dependency on the direct state, and  $\Upsilon : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$ . These two operators define a generalized eigenproblem formed from the direct state.

Define the pair  $(\mathbf{p}, \mathbf{q})$ , called direct and adjoint eigenvectors, associated with the leading eigenvalue<sup>1</sup>  $\sigma \in \mathbb{C}$ , as the solution of the direct and adjoint eigenproblem:

$$\mathbf{p} \in \mathcal{W}_0^2 \quad \text{s.t.} \quad \forall \mathbf{w} \in \mathcal{W}_0 : \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{w}) = \sigma \Upsilon(\mathbf{p}, \mathbf{w}) \quad (3.2a)$$

$$\mathbf{q} \in \mathcal{W}_0^2 \quad \text{s.t.} \quad \forall \mathbf{w} \in \mathcal{W}_0 : \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{w}, \mathbf{q}) = \sigma \Upsilon(\mathbf{w}, \mathbf{q}) \quad (3.2b)$$

We are interested in optimizing the growth rate of the leading eigenvalue  $\Re(\sigma)$  with respect to the parameter  $\beta$ , a parameter that appears in the eigenproblem but not in the direct equation. For this purpose, we wish to evaluate the gradient  $d_\beta \Re(\sigma)$ .

From the definition of the direct and adjoint eigenvectors (3.2), the eigenvalue can be expressed as  $\sigma = \frac{\Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})}$ . Differentiating this expression with respect to the parameter  $\beta$  leads to an expression for the derivative of the growth rate:

$$\begin{aligned} d_\beta \Re(\sigma) &= d_\beta \Re \left( \frac{\Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) \\ &= \Re \left( \frac{\partial_\beta \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q}) + \Gamma_{\alpha,\beta}[\mathbf{x}](d_\beta \mathbf{p}, \mathbf{q}) + \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, d_\beta \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) \\ &\quad - \Re \left( \frac{\Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q}) \cdot (\Upsilon(d_\beta \mathbf{p}, \mathbf{q}) + \Upsilon(\mathbf{p}, d_\beta \mathbf{q}))}{\Upsilon(\mathbf{p}, \mathbf{q})^2} \right) \end{aligned} \quad (3.3)$$

---

<sup>1</sup>For simplicity, we will assume that the leading eigenvalue is isolated and simple.

Using again the expression of the eigenvalue  $\sigma = \frac{\Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p},\mathbf{q})}{\Upsilon(\mathbf{p},\mathbf{q})}$  simplifies the expression:

$$\begin{aligned} d_\beta \Re(\sigma) &= \Re \left( \frac{\partial_\beta \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p},\mathbf{q})}{\Upsilon(\mathbf{p},\mathbf{q})} \right) \\ &+ \Re \left( \frac{(\Gamma_{\alpha,\beta}[\mathbf{x}](d_\beta \mathbf{p},\mathbf{q}) - \sigma \Upsilon(d_\beta \mathbf{p},\mathbf{q})) + (\Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p},d_\beta \mathbf{q}) - \sigma \Upsilon(\mathbf{p},d_\beta \mathbf{q}))}{\Upsilon(\mathbf{p},\mathbf{q})} \right) \end{aligned} \quad (3.4)$$

As the derivative of the direct and adjoint eigenvectors  $d_\beta \mathbf{p}$  and  $d_\beta \mathbf{q}$  belong to the space  $\mathcal{W}_0$ , the definitions of the direct and adjoint eigenvectors (3.2) state that the second term is null:

<b><math>\beta</math>-derivative of the growth rate</b>
$d_\beta \Re(\sigma) = \Re \left( \frac{\partial_\beta \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p},\mathbf{q})}{\Upsilon(\mathbf{p},\mathbf{q})} \right) \quad (3.5)$

Assuming that the analytical expression of the functionals  $\mathcal{N}_\alpha$ ,  $\Gamma_{\alpha,\beta}[\mathbf{x}]$ ,  $\Upsilon$ , and  $\partial_\beta \Gamma_{\alpha,\beta}[\mathbf{x}]$  are known, the gradient of the growth rate  $d_\beta \Re(\sigma)$  can be evaluated by solving the direct equation (3.1), solving the direct and adjoint eigenproblems (3.2) and forming the expression (3.5). This process, schematized in Figure 3.1, is often referred to as modal sensitivity analysis. With this method, it is possible to compute the gradient with respect to parameters that have been introduced for the eigenproblems, such as spanwise wave number or base flow modification. However, it is not possible to compute the gradient with respect to parameters that are present in both the direct equation and the eigenproblems, such as the Reynolds number.

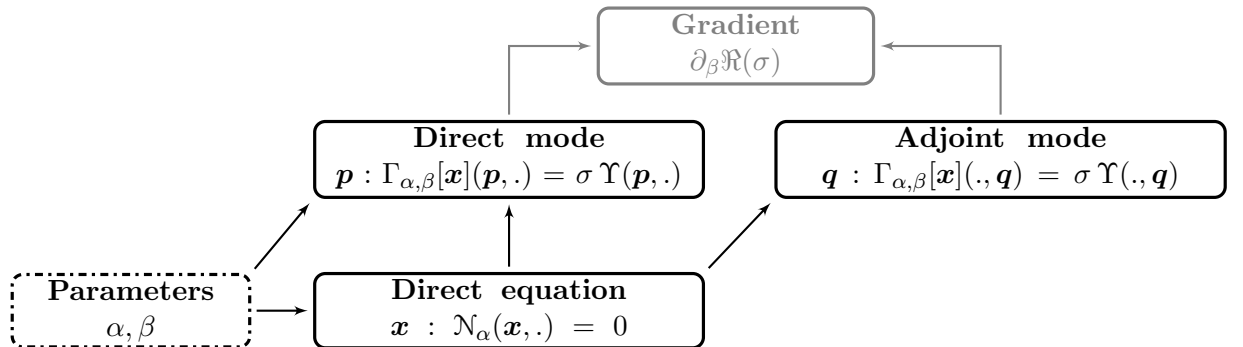


Figure 3.1: Diagram of the modal sensitivity analysis

### 3. Double-decker adjoint algorithm

---

#### 3.1.2 Application

The stability problem associated with the backward-facing slope problem presented in Section 1.3 is three-dimensional. To be able to solve it on a two-dimensional mesh, we introduce a spanwise wave number  $\beta$  and look for unstable mode of the form :  $(\mathbf{w} \cos(\beta z), w_z \sin(\beta z), w_p \cos(\beta z))^t$ . The resulting stability problem belong to the space  $\mathcal{W} = H^1(\Omega)^2 \times H^1(\Omega) \times L^2(\Omega)$  and is formed with the two functionals:

$$\begin{aligned} \Gamma_{\alpha,\beta} [(\mathbf{u}, u_p)] : ((\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto & \int_{\Omega} \nabla \mathbf{w} : \left( \boldsymbol{\eta} \otimes \mathbf{u} + \frac{\nabla \boldsymbol{\eta}}{\alpha} \right) + \nabla w_z \cdot \left( \eta_z \mathbf{u} + \frac{\nabla \eta_z}{\alpha} \right) \\ & + \nabla \mathbf{u} : \boldsymbol{\eta} \otimes \mathbf{w} + \frac{\beta^2}{\alpha} (\mathbf{w} \cdot \boldsymbol{\eta} + w_z \eta_z) - w_p (\operatorname{div} \boldsymbol{\eta} - \beta \eta_z) + \eta_p (\operatorname{div} \mathbf{w} - \beta w_z) \end{aligned} \quad (3.6)$$

and

$$\Upsilon : ((\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto - \int_{\Omega} \mathbf{w} \cdot \boldsymbol{\eta} + w_z \eta_z \quad (3.7)$$

Taking the derivative of  $\Gamma_{\alpha,\beta} [(\mathbf{u}, u_p)]$  with respect to  $\beta$  is straightforward:

$$\partial_{\beta} \Gamma_{\alpha,\beta} [(\mathbf{u}, u_p)] : ((\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto \int_{\Omega} \frac{2\beta}{\alpha} (\mathbf{w} \cdot \boldsymbol{\eta} + w_z \eta_z) + w_p \eta_z - \eta_p w_z \quad (3.8)$$

Using these definitions and previous ones, the modal sensitivity analysis can be implemented to compute the gradient  $d_{\beta} \Re(\sigma)$ . As seen in Figure 3.2, an excellent agreement is reached between the values obtained using the adjoint method and the ones approximated by finite differences.

## 3.2 Parameter optimization

### 3.2.1 Derivation

Consider again the functionals  $\mathcal{N}_{\alpha} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ ,  $\Gamma_{\alpha,\beta}[\mathbf{x}] : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$  and  $\Upsilon : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$  that define the direct state,  $\mathbf{x}$ , solution of the direct equation (3.1) and the direct and adjoint eigenvectors,  $\mathbf{p}$  and  $\mathbf{q}$ , solutions of the eigenproblems (3.2).

We are now interested in optimizing the growth rate of the leading eigenvalue  $\Re(\sigma)$  with respect to the parameter  $\alpha$  and wish therefore to evaluate the gradient  $d_{\alpha} \Re(\sigma)$ . Let introduce the trilinear<sup>1</sup> form  $\Lambda_{\alpha,\beta}[\mathbf{x}] : \mathcal{V} \times \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$  defined by<sup>2</sup>  $\forall \mathbf{v} \in \mathcal{V}$  :

---

<sup>1</sup>Linear in each of its three arguments.

<sup>2</sup> $\partial_{\cdot}$  denotes the partial derivative with respect to the argument inside the brackets.

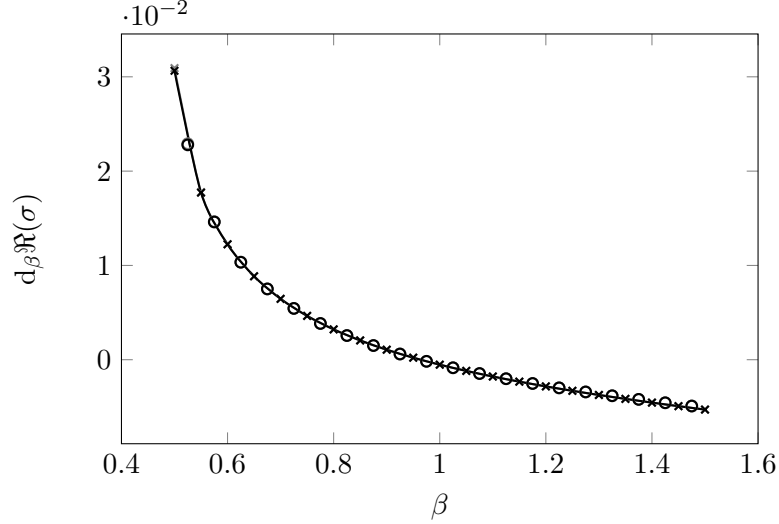


Figure 3.2: Validation of the formula of the  $\beta$ -parameter gradient of the growth rate (3.5) using both a coarse mesh (gray) and a fine mesh (black). The circles corresponds to the gradient obtained with finite differences while the crosses and the fitting spline represents the gradient computed using the adjoint method.

$\Lambda_{\alpha,\beta}[\mathbf{v}] = \partial \Gamma_{\alpha,\beta}[\cdot](\mathbf{v})$ . Repeating the step (3.3–3.5) for  $d_\alpha \Re(\sigma)$  results in the expression:

$$d_\alpha \Re(\sigma) = \Re \left( \frac{\partial_\alpha \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} + \frac{\Lambda_{\alpha,\beta}[\mathbf{x}](d_\alpha \mathbf{x}, \mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) \quad (3.9)$$

The second term of this expression contains the derivative of the direct state  $d_\alpha \mathbf{x}$ . We already know from Section 2.1 that the derivative of the direct state  $d_\alpha \mathbf{x}$  is solution of the equation:

$$d_\alpha \mathbf{x} \in \mathcal{V}_0 \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \partial_\alpha \mathcal{N}_\alpha(\mathbf{x}, \mathbf{v}) + \mathcal{L}_\alpha[\mathbf{x}](d_\alpha \mathbf{x}, \mathbf{v}) = 0 \quad (3.10)$$

Define now the adjoint state  $\mathbf{y}$  as the solution of the adjoint equation<sup>1</sup>:

$$\mathbf{y} \in \mathcal{V}_0^2 \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_\alpha[\mathbf{x}](\mathbf{v}, \mathbf{y}) = \Lambda_{\alpha,\beta}[\mathbf{x}](\mathbf{v}, \mathbf{p}, \mathbf{q}) \quad (3.11)$$

As  $d_\alpha \mathbf{x} \in \mathcal{V}_0$ , the adjoint equation (3.11) leads to the following relation:

$$d_\alpha \Re(\sigma) = \Re \left( \frac{\partial_\alpha \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) + \Re \left( \frac{\mathcal{L}_\alpha[\mathbf{x}](d_\alpha \mathbf{x}, \mathbf{y})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) \quad (3.12)$$

<sup>1</sup>Because the eigenvectors  $\mathbf{p}$  and  $\mathbf{q}$  can have complex values, this equation must be interpreted as an equality on the complex field.

### 3. Double-decker adjoint algorithm

Similarly, as  $\mathbf{y} \in \mathcal{V}_0$ , the equation (3.10) leads to the final expression of the gradient  $d_\alpha \mathfrak{R}(\sigma)$ :

**$\alpha$ -derivative of the growth rate**

$$d_\alpha \mathfrak{R}(\sigma) = \Re \left( \frac{\partial_\alpha \Gamma_{\alpha,\beta}[\mathbf{x}](\mathbf{p}, \mathbf{q})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) - \Re \left( \frac{\partial_\alpha \mathcal{N}_\alpha(\mathbf{x}, \mathbf{y})}{\Upsilon(\mathbf{p}, \mathbf{q})} \right) \quad (3.13)$$

Assuming that the analytical expression of the functionals  $\mathcal{N}_\alpha$ ,  $\Gamma_{\alpha,\beta}[\mathbf{x}]$ ,  $\Upsilon$ ,  $\Lambda_{\alpha,\beta}[\mathbf{x}]$ ,  $\partial_\alpha \Gamma_{\alpha,\beta}[\mathbf{x}]$  and  $\partial_\alpha \mathcal{N}_\alpha$  are known, the gradient of the growth rate  $d_\alpha \mathfrak{R}(\sigma)$  can be evaluated by solving the direct equation (3.1), solving the direct and adjoint eigenproblems (3.2), solving the adjoint equation (3.11) and forming the expression (3.13). This process, schematized in Figure 3.3, is simply a combination of the standard adjoint algorithm of Chapter 2 with the modal sensitivity analysis of Section 3.1.

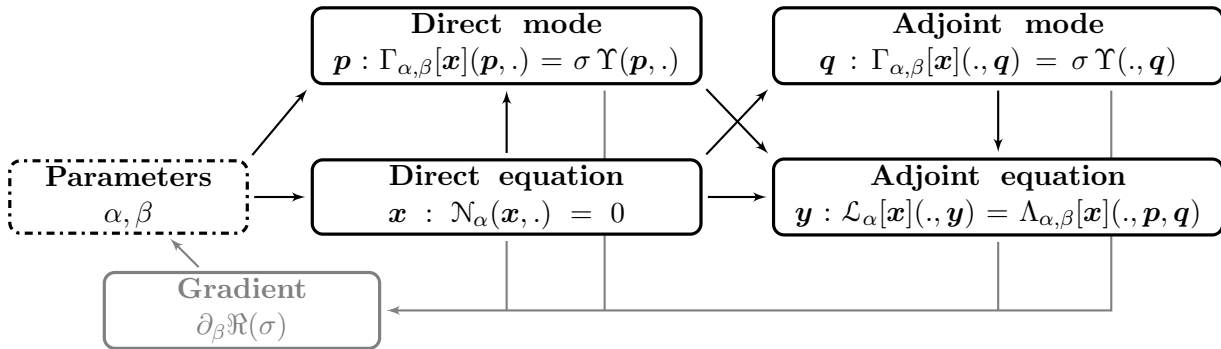


Figure 3.3: Diagram of the double-decker adjoint algorithm

### 3.2.2 Application

Going back to the problem introduced in Section 1.3, the only functionals that we do not yet know are  $\partial_\alpha \Gamma_{\alpha,\beta}[(\mathbf{u}, u_p)]$  and  $\Lambda_{\alpha,\beta}[(\mathbf{u}, u_p)]$ . Differentiating  $\Gamma_{\alpha,\beta}[(\mathbf{u}, u_p)]$  with respect to  $\alpha$  leads to:

$$\begin{aligned} \partial_\alpha \Gamma_{\alpha,\beta}[(\mathbf{u}, u_p)] : ((\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto & -\frac{1}{\alpha^2} \int_{\Omega} \nabla \mathbf{w} : \nabla \boldsymbol{\eta} + \nabla w_z \cdot \nabla \eta_z \\ & + \beta^2 (\mathbf{w} \cdot \boldsymbol{\eta} + w_z \eta_z) \quad (3.14) \end{aligned}$$

---

Similarly, differentiating  $\Gamma_{\alpha,\beta} [(\mathbf{u}, u_p)]$  with respect to  $(\mathbf{u}, u_p)$  leads to:

$$\Lambda_{\alpha,\beta} [(\mathbf{u}, u_p)] : ((\mathbf{v}, v_p), (\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto \int_{\Omega} \nabla \mathbf{w} : \boldsymbol{\eta} \otimes \mathbf{v} + \eta_z (\nabla w_z \cdot \mathbf{v}) + \nabla \mathbf{v} : \boldsymbol{\eta} \otimes \mathbf{w} \quad (3.15)$$

Since  $\Gamma_{\alpha,\beta} [(\mathbf{u}, u_p)]$  has a linear dependence on  $(\mathbf{u}, u_p)$ ,  $\Lambda_{\alpha,\beta} [(\mathbf{u}, u_p)]$  does not depend on the direct state.

The implementation of the double-decker adjoint algorithm to compute the gradient  $d_{\alpha} \mathfrak{R}(\sigma)$  clearly validates the method. As shown in Figure 3.4 the values of the gradient obtained by finite differences and using the adjoint method matches.

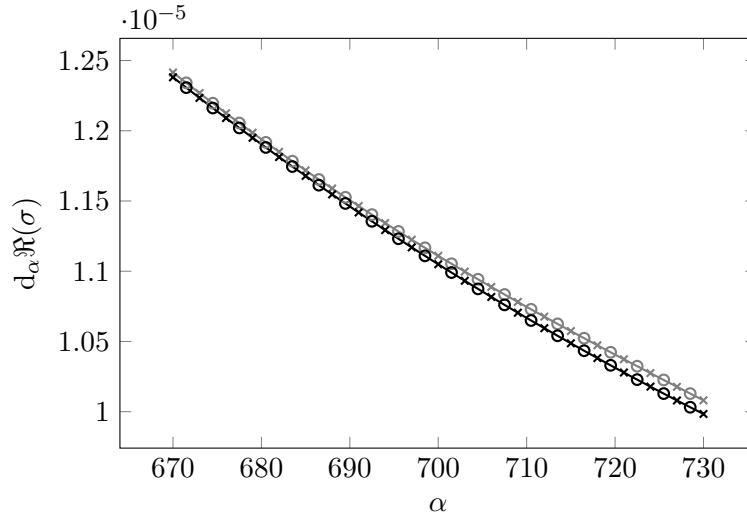


Figure 3.4: Validation of the formula of the  $\alpha$ -parameter gradient of the growth rate (3.13) using both a coarse mesh (gray) and a fine mesh (black). The circles corresponds to the gradient obtained with finite differences while the crosses and the fitting spline represents the gradient computed using the adjoint method.

### 3.3 Shape optimization

#### 3.3.1 Derivation

Consider again the functionals  $\mathcal{N}_{\alpha} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ ,  $\Gamma_{\alpha,\beta}[\mathbf{x}] : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$ ,  $\Upsilon : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$  and  $\Lambda_{\alpha,\beta}[\mathbf{x}] : \mathcal{V} \times \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$ . These functionals are used to define the direct state,  $\mathbf{x}$ , solution of the direct equation (3.1) and the direct and adjoint eigenvectors,  $\mathbf{p}$  and

### 3. Double-decker adjoint algorithm

---

$\mathbf{q}$ , solutions of the eigenproblems (3.2).

The equation satisfied by the shape derivative of the direct state is known from Section 2.2. The shape derivative of the direct state  $d_{\mathbf{h}}\mathbf{x}$  is solution of the equation:

$$d_{\mathbf{h}}\mathbf{x} \in \mathcal{V} \left[ -\mathbb{1}_{\bar{\Gamma}^d}((\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}}\bar{\mathbf{x}}) \right] \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_{\alpha}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{v}) = 0 \quad (3.16)$$

The equation satisfied by the shape derivative of the direct eigenvector  $d_{\mathbf{h}}\mathbf{p}$  can be obtained from the direct eigenproblem (3.2a) using some shape derivative techniques. The shape derivative of the direct eigenvector  $d_{\mathbf{h}}\mathbf{p}$  is solution of the equation:

$$\begin{aligned} d_{\mathbf{h}}\mathbf{p} &\in \mathcal{W} \left[ -\mathbb{1}_{\bar{\Gamma}^d}((\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}}\bar{\mathbf{p}}) \right] \quad \text{s.t.} \\ \forall \mathbf{w} \in \mathcal{W}_0 : \Lambda_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{p}, \mathbf{w}) + \Gamma_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{p}, \mathbf{w}) &= \sigma \Upsilon(d_{\mathbf{h}}\mathbf{p}, \mathbf{w}) + (d_{\mathbf{h}}\sigma) \Upsilon(\mathbf{p}, \mathbf{w}) \end{aligned} \quad (3.17)$$

As the adjoint eigenvector  $\mathbf{q}$  belongs to  $\mathcal{W}_0$ , the shape derivative of the eigenvalue is:

$$d_{\mathbf{h}}\sigma = \frac{1}{\Upsilon(\mathbf{p}, \mathbf{q})} \left( \Lambda_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{p}, \mathbf{q}) + \Gamma_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{p}, \mathbf{q}) - \sigma \Upsilon(d_{\mathbf{h}}\mathbf{p}, \mathbf{q}) \right) \quad (3.18)$$

Define the adjoint state  $\mathbf{y}$  as the solution of the adjoint equation<sup>1</sup>:

$$\mathbf{y} \in \mathcal{V}_0^2 \quad \text{s.t.} \quad \forall \mathbf{v} \in \mathcal{V}_0 : \mathcal{L}_{\alpha}[\mathbf{x}](\mathbf{v}, \mathbf{y}) = \Lambda_{\alpha,\beta}[\mathbf{x}](\mathbf{v}, \mathbf{p}, \mathbf{q}) \quad (3.19)$$

As  $\mathbf{y} \in \mathcal{V}_0$ , equation (3.16) gives in particular that  $\mathcal{L}_{\alpha}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{y}) = 0$ . Because of the definition of the adjoint equation (3.19), the Boundary residual formula for an adjoint equation (A.7) can be applied to the bilinear form  $\mathcal{L}_{\alpha}[\mathbf{x}]$  and the linear form  $\cdot \mapsto \Lambda_{\alpha,\beta}[\mathbf{x}](\cdot, \mathbf{p}, \mathbf{q})$ :

$$\Lambda_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{x}, \mathbf{p}, \mathbf{q}) = \langle \gamma^{\dagger} \{ \mathcal{L}_{\alpha}[\mathbf{x}] \}(\mathbf{y}) - \gamma \{ \Lambda_{\alpha,\beta}[\mathbf{x}](\cdot, \mathbf{p}, \mathbf{q}) \}, (\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}}\bar{\mathbf{x}} \rangle_{L^2(\bar{\Gamma}^d)} \quad (3.20)$$

Moreover because equation (3.2b) is satisfied, the boundary residual formula for an adjoint eigenproblem (A.8) can be applied to the bilinear form  $\Gamma_{\alpha,\beta}[\mathbf{x}] - \sigma \Upsilon$ :

$$\Gamma_{\alpha,\beta}[\mathbf{x}](d_{\mathbf{h}}\mathbf{p}, \mathbf{q}) - \sigma \Upsilon(d_{\mathbf{h}}\mathbf{p}, \mathbf{q}) = -\langle \gamma^{\dagger} \{ \Gamma_{\alpha,\beta}[\mathbf{x}] \}(\mathbf{q}) - \sigma \gamma^{\dagger} \{ \Upsilon \}(\mathbf{q}), (\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}}\bar{\mathbf{p}} \rangle_{L^2(\bar{\Gamma}^d)} \quad (3.21)$$

Gathering the equations (3.18), (3.20) and (3.21) leads to an expression for the

---

<sup>1</sup>Once again, this adjoint equation is identical to the one introduced for the parameter optimization.

shape derivative of the growth rate:

### Shape gradient of the growth rate

$$\begin{aligned} d_{\mathbf{h}}\mathfrak{R}(\sigma) = \Re \left[ \frac{1}{\Upsilon(\mathbf{p}, \mathbf{q})} \left( \langle \gamma^\dagger \{ \mathcal{L}_\alpha[\mathbf{x}] \}(\mathbf{y}) - \gamma \{ \Lambda_{\alpha, \beta}[\mathbf{x}](\cdot, \mathbf{p}, \mathbf{q}) \}, (\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}} \bar{\mathbf{x}} \rangle_{L^2(\bar{\Gamma}^d)} \right. \right. \\ \left. \left. - \langle \gamma^\dagger \{ \Gamma_{\alpha, \beta}[\mathbf{x}] \}(\mathbf{q}) - \sigma \gamma^\dagger \{ \Upsilon \}(\mathbf{q}), (\mathbf{h} \cdot \mathbf{n}) \partial_{\mathbf{n}} \bar{\mathbf{p}} \rangle_{L^2(\bar{\Gamma}^d)} \right) \right] \quad (3.22) \end{aligned}$$

If the analytical expression of the functionals  $\mathcal{L}_\alpha[\mathbf{x}]$ ,  $\Gamma_{\alpha, \beta}[\mathbf{x}]$ ,  $\Upsilon$  and  $\Lambda_{\alpha, \beta}[\mathbf{x}]$  are known, Green's identities must be used to decompose the functionals in an appropriate form. These decompositions will define  $\gamma^\dagger \{ \mathcal{L}_\alpha[\mathbf{x}] \}$ ,  $\gamma^\dagger \{ \Gamma_{\alpha, \beta}[\mathbf{x}] \}$ ,  $\gamma^\dagger \{ \Upsilon \}$  and  $\gamma \{ \Lambda_{\alpha, \beta}[\mathbf{x}](\cdot, \mathbf{p}, \mathbf{q}) \}$ . The gradient  $d_{\mathbf{h}}\mathfrak{R}(\sigma)$  can then be evaluated by solving the direct equation (3.1), solving the direct and adjoint eigenproblems (3.2), solving the adjoint equation (3.19) and forming the expression (3.22).

### 3.3.2 Application

Going back to the problem introduced in Section 1.3, we need now to obtain the expression of the boundary residual from the expression of  $\Gamma_{\alpha, \beta}[(\mathbf{u}, u_p)]$ ,  $\Upsilon$  and  $\Lambda_{\alpha, \beta}[(\mathbf{u}, u_p)]$  using Green's identities.

Using all the Green's identities (A.10):

$$\begin{aligned} \Gamma_{\alpha, \beta}[(\mathbf{u}, u_p)] : ((\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) \mapsto \int_{\Omega} -\mathbf{w} \cdot \operatorname{div} \left( \boldsymbol{\eta} \otimes \mathbf{u} + \frac{\nabla \boldsymbol{\eta}}{\alpha} \right) + \nabla \mathbf{u} : \boldsymbol{\eta} \otimes \mathbf{w} \\ - w_z \operatorname{div} \left( \eta_z \mathbf{u} + \frac{\nabla \eta_z}{\alpha} \right) + \frac{\beta^2}{\alpha} (\mathbf{w} \cdot \boldsymbol{\eta} + w_z \eta_z) - w_p (\operatorname{div} \boldsymbol{\eta} - \beta \eta_z) - \nabla \eta_p \cdot \mathbf{w} - \beta \eta_p w_z \\ + \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n}) (\boldsymbol{\eta} \cdot \mathbf{w} + w_z \eta_z) + \alpha^{-1} (\partial_{\mathbf{n}} \boldsymbol{\eta} \cdot \mathbf{w} + \partial_{\mathbf{n}} \eta_z w_z) + \eta_p (\mathbf{w} \cdot \mathbf{n}) \quad (3.23) \end{aligned}$$

This form matches the decomposition (A.3b) so by definition:

$$\gamma^\dagger \{ \Gamma_{\alpha, \beta}[(\mathbf{u}, u_p)] \} ((\boldsymbol{\eta}, \eta_z, \eta_p)) = \begin{pmatrix} (\mathbf{u} \cdot \mathbf{n}) \boldsymbol{\eta} + \alpha^{-1} \partial_{\mathbf{n}} \boldsymbol{\eta} + \eta_p \mathbf{n} \\ (\mathbf{u} \cdot \mathbf{n}) \eta_z + \alpha^{-1} \partial_{\mathbf{n}} \eta_z \end{pmatrix} = \begin{pmatrix} \alpha^{-1} \partial_{\mathbf{n}} \boldsymbol{\eta} + \eta_p \mathbf{n} \\ \alpha^{-1} \partial_{\mathbf{n}} \eta_z \end{pmatrix} \quad (3.24)$$

with the last equality due to the zero Dirichlet boundary condition of the velocity of the direct state  $\mathbf{u}$  on  $\Gamma^d$ .

### 3. Double-decker adjoint algorithm

---

$\Upsilon$  is already in the correct form (A.3b) so we have:

$$\gamma^\dagger \{ \Upsilon \} ((\boldsymbol{\eta}, \eta_z, \eta_p)) = \mathbf{0} \quad (3.25)$$

Using the Green's identity (A.10c):

$$\begin{aligned} \Lambda_{\alpha,\beta} [(\mathbf{u}, u_p)] ((\mathbf{v}, v_p), (\mathbf{w}, w_z, w_p), (\boldsymbol{\eta}, \eta_z, \eta_p)) &= \int_{\Omega} \nabla \mathbf{w} : \boldsymbol{\eta} \otimes \mathbf{v} + \eta_z (\nabla w_z \cdot \mathbf{v}) \\ &\quad - \int_{\Omega} \mathbf{v} \cdot \operatorname{div}(\boldsymbol{\eta} \otimes \mathbf{w}) + \int_{\Gamma} (\mathbf{v} \cdot \boldsymbol{\eta})(\mathbf{w} \cdot \mathbf{n}) \end{aligned} \quad (3.26)$$

This form matches the required decomposition (A.3b) so if we write  $(\hat{\mathbf{u}}, \hat{u}_z, \hat{u}_p)$  and  $(\check{\mathbf{u}}, \check{u}_z, \check{u}_p)$  the direct and adjoint eigenvectors:

$$\gamma \{ \Lambda_{\alpha,\beta} [(\mathbf{u}, u_p)] (\cdot, (\hat{\mathbf{u}}, \hat{u}_z, \hat{u}_p), (\check{\mathbf{u}}, \check{u}_z, \check{u}_p)) \} = (\hat{\mathbf{u}} \cdot \mathbf{n}) \check{\mathbf{u}} = 0 \quad (3.27)$$

with the last equality due to the zero Dirichlet boundary condition of the velocity of the direct eigenvector  $(\hat{\mathbf{u}}, \hat{u}_z)$  on  $\Gamma^d$ .

Noting  $(\bar{\mathbf{u}}, \bar{u}_p)$  the direct state and  $(\tilde{\mathbf{u}}, \tilde{u}_p)$  the adjoint state, we finally obtain an explicit form for the shape derivative:

$$\begin{aligned} d_s \mathfrak{R}(\sigma) &= \\ \mathfrak{R} \left[ \frac{\langle \alpha^{-1}(\partial_n \bar{\mathbf{u}} \cdot \partial_n \tilde{\mathbf{u}} - \partial_n \hat{\mathbf{u}} \cdot \partial_n \check{\mathbf{u}} - \partial_n \hat{u}_z \partial_n \check{u}_z) + \tilde{u}_p \partial_n \bar{\mathbf{u}} \cdot \mathbf{n}, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \rangle_{L^2(\bar{\Gamma}^d)}}{\langle \hat{\mathbf{u}}, \check{\mathbf{u}} \rangle_{L^2(\Omega)} + \langle \hat{u}_z, \check{u}_z \rangle_{L^2(\Omega)}} \right]. \end{aligned} \quad (3.28)$$

Once again, we can simplify this expression using the relation  $\partial_n \bar{\mathbf{u}} \cdot \mathbf{n} = 0$ :

$$d_s \mathfrak{R}(\sigma) = \frac{1}{\alpha} \mathfrak{R} \left[ \frac{\langle \partial_n \bar{\mathbf{u}} \cdot \partial_n \tilde{\mathbf{u}} - \partial_n \hat{\mathbf{u}} \cdot \partial_n \check{\mathbf{u}} - \partial_n \hat{u}_z \partial_n \check{u}_z, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \rangle_{L^2(\bar{\Gamma}^d)}}{\langle \hat{\mathbf{u}}, \check{\mathbf{u}} \rangle_{L^2(\Omega)} + \langle \hat{u}_z, \check{u}_z \rangle_{L^2(\Omega)}} \right]. \quad (3.29)$$

As before, we may prefer a slightly different expression of the same quantity:

$$d_s \mathfrak{R}(\sigma) = \frac{1}{\alpha} \mathfrak{R} \left[ \frac{\langle \nabla \bar{\mathbf{u}} : \nabla \tilde{\mathbf{u}} - \nabla \hat{\mathbf{u}} : \nabla \check{\mathbf{u}} - \nabla \hat{u}_z \cdot \nabla \check{u}_z, \mathbf{h}_{\mathcal{B}(s)} \cdot \mathbf{n}_{\mathcal{B}(s)} \rangle_{L^2(\bar{\Gamma}^d)}}{\langle \hat{\mathbf{u}}, \check{\mathbf{u}} \rangle_{L^2(\Omega)} + \langle \hat{u}_z, \check{u}_z \rangle_{L^2(\Omega)}} \right]. \quad (3.30)$$

The implementation of this formula results in the values of the gradient  $d_s \mathfrak{R}(\sigma)$

---

displayed in Figure 3.5. With the fine mesh, a good agreement is reached between the finite difference approximation and the adjoint method. With the coarse mesh, the finite difference approximation is scattered but the adjoint method produces surprisingly good values.

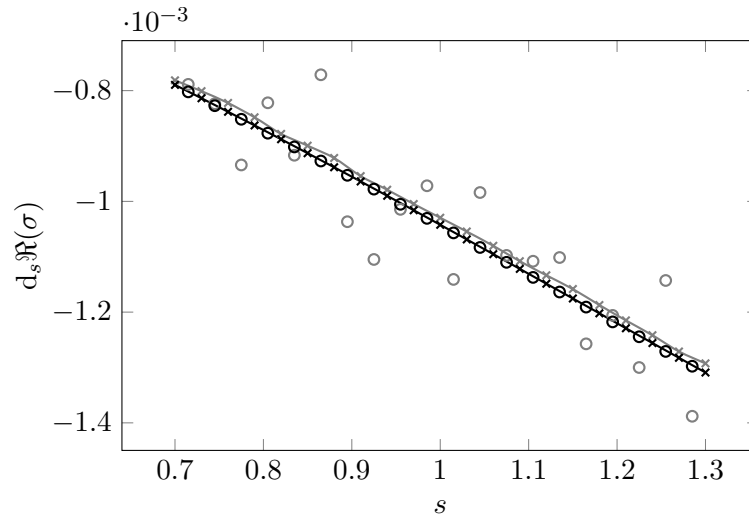


Figure 3.5: Validation of the formula of the shape gradient of the growth rate (3.22) using both a coarse mesh (gray) and a fine mesh (black). The circles corresponds to the gradient obtained with finite differences while the crosses and the fitting spline represents the gradient computed using the adjoint method.



## Chapter 4

# Future Work

In the short term, the priority is to implement some automatic deformation of the mesh so that the shape gradient can be used to deform the backward facing slope and find the locally optimal shape, for functional cost function (standard algorithm) as well as cost function based on the leading growth rate of the linearized operator (double-decker algorithm).

At the same time, the mathematical framework will undergo some refinements : a few mathematical points, such as the required regularity of the domain, need clarification and the standard algorithm will be modified to incorporate cost functions defined on the boundary<sup>1</sup>.

The medium-term goal is to apply these techniques to a few relevant problems of limited complexity. One of them could be an oscillator flow with the objective to minimize the energy lost while keeping the flow stable. It could also be interesting to work on an amplifier flow and try to minimize the energy of the optimal linear response to a forcing. Indeed this problem can be written as an eigenvalue problem involving the resolvent [27, chap. 4].

It may be necessary at this stage to improve the shape optimization algorithm from a gradient descent method to a quasi-newton method. Such methods, called Sobolev smoothing, have been designed for shape optimization and speed up the convergence while dealing with regularity problem of the boundary [5].

In a longer term, several extensions of the concept are possible:

- Extend the double-decker algorithm to systems where the primary problem and/or

---

<sup>1</sup>Lift and drag fall in this category.

## 4. Future Work

---

the auxiliary problem are an evolution equation with a finite time. It will then be possible to treat a much wider range of problems. This extension should be theoretically manageable but more difficult on the computational side as it requires to implement adjoint-looping techniques [16].

- Extend both the standard and double-decker algorithm to allow topological change of the domain. The present algorithm can continuously deform the shape but cannot generate or annihilate holes in the domain. The concept of topological shape derivative was introduced by Sokolowski & Zochowski [31] and was since primarily applied to structural mechanics [1]. Most of the actual application relies on level-set method but a more straightforward computation of the topological shape derivative may be simpler for fluid flows.
- Extend the double-decker algorithm to system where the auxiliary problem is a control related problem. Typically, given an actuator and a sensor model, we want to find the best shape of the domain and the best location of the sensor and actuator so that the linear control is the most efficient. A computational problem may arise as optimal  $\mathcal{H}_2$  or  $\mathcal{H}_\infty$  controllers are usually found using Ricatti equations or Linear Matrix Inequalities (LMI) which are not tractable for large systems. Nevertheless, it may be possible to reduce the system size before designing the controller [20] or to solve for simpler problem that gives bounds on the efficiency of the controller without building the controller [33].
- Extend both the standard and double-decker algorithms to uncertain systems, i.e. systems in which some parameters do not have a fixed value but are random variable following a specified probability distribution. This type of problems, often called robust shape optimization, have already been investigated [18, 26]. The Monte–Carlo method would probably be adapted to problems with a couple of random variable as it is easy to implement [8]. Another option would be to use Stochastic Finite Element Methods that add dimensions to the simulation to solve for the probability distribution of the variables [7]. Although more precise, this method could be computationally expensive.

All these improvements takes some work and adds nothing radically new. To my mind, they should only be considered if a well-defined and engaging problem<sup>1</sup> requires one of this method. If not, it may be worth to turn to a more ambitious project.

---

<sup>1</sup>Such problem could emerge from the European collaborative project ANADE.

---

A first path would be to study how noise influence transition and if successful try to control these transitions with shape optimization. While transition in oscillators flows are accurately described by modal stability analysis, the transition mechanism in amplifier flow is different. Noise is linearly amplified to a point where it triggers a jump to another orbit of the Navier–Stokes equation. Although it may be possible to deal with this system by introducing the forcing or the initial condition that induce the largest linear growth, this is not completely satisfactory as noise obeys to some temporal and spatial properties and cannot be captured by an initial condition or a forcing. However, modeling noise in dynamical system is not an easy task as it requires to immerse the Navier–Stokes equations in the thorny world of stochastic partial differential equation. Fortunately, these equations are at the core of mathematical finance so many mathematicians have looked into it and a few algorithms have been designed to simulate them. In particular, Bismut [2] has introduced a backward stochastic differential equation that presents similarities with adjoint equations [17]. Although this path could be mathematically challenging and computationally demanding, it would, if successful, bring a more complete view of shape optimization of transitions in dynamical system.

Another route, more oriented towards industrial application, would be to work on shape optimization of turbulent flow. Turbulent flows are at the center of many industrial systems but are also both theoretically and numerically puzzling. Shape optimization of turbulent flows has already been attempted [22, 32]. This attempts use either Reynolds Averaged Navier–Stokes (RANS) or Large Eddy Simulation (LES) turbulence model.

Trying to apply the standard and double-decker algorithm on a turbulent flow would present some technical challenges. The deepest concern is about the accuracy of the turbulence models. Most RANS model have a strong dependence on the geometry and perform badly when the various coefficients have not been adjusted to the geometry. As we seek to repeatedly modify the shape of our domain, we need a model that perform well in a wide range of geometry. LES model are less influenced by a change in the geometry but their implementation with Finite Element Method is delicate. An alternative could be to use Implicit LES turbulence models which are much easier to implement with Finite Element Method as the turbulence model is included in the numerical scheme[21]. However, wall functions are still used to avoid resolving the boundary layer which may be problematic for shape optimization.

Yet, the main issues with stability of turbulent flows are more theoretical than technical. While the stability of a steady solution has a clear mathematical and physical meaning,

#### 4. Future Work

---

the stability of a spatially and temporally averaged solution of a chaotic system is a questionable concept. Any instantaneous snapshot of the flow would be unstable and it is not clear how the averaging process affect the stability. Beside, whereas linear mechanisms are at the core of the growth of disturbances evolving on top of a steady solution, the same cannot be said for disturbances evolving on top of an averaged flow [11]. If the solution is found to be unstable, it may not be truly relevant as the averaged system may transition to a state very close from the initial state.

Therefore, it would be preferable to replace the eigenproblem of the stability equation by the non-linear evolution equation of a perturbation that grows on top of the averaged flow. It would have more prediction capabilities than the stability analysis, would make more sense in my opinion but would require to implement an adjoint-looping technique mentioned above.

# Appendix A

## Mathematical tools

### A.1 Boundary residual

This section establishes some mathematical results required to derive the shape derivative formulae (2.16) and (3.22). Two lemmata are first stated to condense the following discussion.

#### Zero Dirichlet $L^2$ -approximation lemma

$$\forall z \in H^k(\Omega), \exists \{z_n\}_{n \in \mathbb{N}} \in H_0^k(\Omega)^{\mathbb{N}} : \lim_{n \rightarrow \infty} \|z - z_n\|_{L^2(\Omega)} = 0 \quad (\text{A.1})$$

The proof of this lemma can be carried out by explicitly building the sequence  $\{z_n\}_{n \in \mathbb{N}}$  from  $z$ . However this proof requires to introduce mollifiers [6, p. 713] which is quite tedious. The difficulty can be bypassed if we admit the density of  $C_0^\infty(\Omega)$  in  $L^2(\Omega)$ .

*Proof.* Consider  $z \in H^k(\Omega)$ . As  $H^k(\Omega) \subset L^2(\Omega)$  and  $\overline{C_0^\infty(\Omega)}^{\|\cdot\|_{L^2(\Omega)}} = L^2(\Omega)$ , we get using the sequential definition of the density that  $\exists \{z_n\}_{n \in \mathbb{N}} \in C_0^\infty(\Omega)^{\mathbb{N}} \subset H_0^k(\Omega)^{\mathbb{N}}$  such that  $\lim_{n \rightarrow \infty} \|z - z_n\|_{L^2(\Omega)} = 0$ .  $\square$

#### Extended zero Dirichlet $L^2$ -approximation lemma

$$\forall z \in \bar{\mathfrak{S}}, \exists \{z_n\}_{n \in \mathbb{N}} \in \bar{\mathfrak{S}}_0^{\mathbb{N}} : \begin{cases} \lim_{n \rightarrow \infty} \|z - z_n\|_{L^2(\Omega)} = 0 \\ \lim_{n \rightarrow \infty} \|z \upharpoonright_{\Gamma^\sigma} - z_n \upharpoonright_{\Gamma^\sigma}\|_{L^2(\Gamma^\sigma)} = 0 \end{cases} \quad (\text{A.2})$$

## A. Mathematical tools

---

The proof of this second lemma relies only on the first lemma (A.1) and on the properties of the trace operator  $\cdot|_{\Gamma}$ .

*Proof.* Consider  $\mathbf{z} \in \bar{\mathcal{S}}$ . As the trace  $\cdot|_{\Gamma} : \bar{\mathcal{S}} \rightarrow H^{\frac{1}{2}}(\Gamma)^{\bar{k}}$  is a surjective application, we can find  $\mathbf{z}_{\gamma} \in \bar{\mathcal{S}}_0$  such that  $\mathbf{z} - \mathbf{z}_{\gamma} \in \{\mathbf{s} \in \bar{\mathcal{S}} \mid \mathbf{s}|_{\Gamma^{\sigma}} = \mathbf{0}\}$ . By applying the lemma (A.1) to each components of  $\mathbf{z} - \mathbf{z}_{\gamma}$ , which belong to  $H^1(\Omega)$  by definition of  $\bar{\mathcal{S}}$ , we get that  $\exists \{\mathbf{z}_n\}_{n \in \mathbb{N}} \in \{\mathbf{s} \in \mathcal{S} \mid \mathbf{s}|_{\Gamma} = \mathbf{0}\}^{\mathbb{N}}$  such that  $\lim_{n \rightarrow \infty} \|\mathbf{z} - \mathbf{z}_{\gamma} - \mathbf{z}_n\|_{L^2(\Omega)} = 0$ . Moreover, considering the spaces in which  $\mathbf{z} - \mathbf{z}_{\gamma}$  and the sequence  $\{\mathbf{z}_n\}_{n \in \mathbb{N}}$  belong, we have  $\forall n \in \mathbb{N} : \|\mathbf{z}|_{\Gamma^{\sigma}} - \mathbf{z}_{\gamma}|_{\Gamma^{\sigma}} - \mathbf{z}_n|_{\Gamma^{\sigma}}\|_{L^2(\Gamma^{\sigma})} = \|\mathbf{z}|_{\Gamma^{\sigma}} - \mathbf{z}_{\gamma}|_{\Gamma^{\sigma}}\|_{L^2(\Gamma^{\sigma})} = 0$ . Therefore the sequence  $\{\mathbf{z}_n + \mathbf{z}_{\gamma}\}_{n \in \mathbb{N}} \in \bar{\mathcal{S}}_0^{\mathbb{N}}$  meet the desired requirements.  $\square$

Consider  $a : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$  a bilinear form and  $b : \mathcal{S} \rightarrow \mathbb{R}$  a linear form. Let assume than these forms can be decomposed in the following manner:

$$\forall \mathbf{s} = (\bar{\mathbf{s}}, \dot{\mathbf{s}}) \in \mathcal{S} : b(\mathbf{s}) = \dot{B}(\dot{\mathbf{s}}) + \langle B_{\Omega}, \bar{\mathbf{s}} \rangle_{L^2(\Omega)} + \langle B_{\Gamma^{\sigma}}, \bar{\mathbf{s}}|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} + \langle \gamma\{b\}, \bar{\mathbf{s}}|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} \quad (\text{A.3a})$$

$$\forall \mathbf{s}_1 = (\bar{\mathbf{s}}_1, \dot{\mathbf{s}}_1) \in \mathcal{S}, \forall \mathbf{s}_2 \in \mathcal{S}^2 : a(\mathbf{s}_1, \mathbf{s}_2) = \dot{A}(\dot{\mathbf{s}}_1, \dot{\mathbf{s}}_2) + \langle A_{\Omega}^{\dagger}(\mathbf{s}_2), \bar{\mathbf{s}}_1 \rangle_{L^2(\Omega)} + \langle A_{\Gamma^{\sigma}}^{\dagger}(\mathbf{s}_2), \bar{\mathbf{s}}_1|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} + \langle \gamma^{\dagger}\{a\}(\mathbf{s}_2), \bar{\mathbf{s}}_1|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} \quad (\text{A.3b})$$

Define  $\mathbf{f} \in \mathcal{S}_0^2$  as the solution of the adjoint equation  $\forall \mathbf{s} \in \mathcal{S}_0 : a(\mathbf{s}, \mathbf{f}) = b(\mathbf{s})$ . Using the decomposition (A.3) and the definition of the space  $\mathcal{S}_0$ , the integral on the boundary portion  $\Gamma^d$  cancels and leads to the equality:

$$\forall \mathbf{s} \in \mathcal{S}_0 : \dot{A}(\dot{\mathbf{s}}, \mathbf{f}) + \langle A_{\Omega}^{\dagger}(\mathbf{f}), \bar{\mathbf{s}} \rangle_{L^2(\Omega)} + \langle A_{\Gamma^{\sigma}}^{\dagger}(\mathbf{f}), \bar{\mathbf{s}}|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} = \dot{B}(\dot{\mathbf{s}}) + \langle B_{\Omega}, \bar{\mathbf{s}} \rangle_{L^2(\Omega)} + \langle B_{\Gamma^{\sigma}}, \bar{\mathbf{s}}|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} \quad (\text{A.4})$$

Thanks to the lemma (A.2) and to the Cauchy–Schwarz inequality, the equation (A.4) can be prolonged to  $\mathcal{S}$  :

$$\forall \mathbf{e} \in \mathcal{S} : \dot{A}(\dot{\mathbf{e}}, \mathbf{f}) + \langle A_{\Omega}^{\dagger}(\mathbf{f}), \bar{\mathbf{e}} \rangle_{L^2(\Omega)} + \langle A_{\Gamma^{\sigma}}^{\dagger}(\mathbf{f}), \bar{\mathbf{e}}|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} = \dot{B}(\dot{\mathbf{e}}) + \langle B_{\Omega}, \bar{\mathbf{e}} \rangle_{L^2(\Omega)} + \langle B_{\Gamma^{\sigma}}, \bar{\mathbf{e}}|_{\Gamma^{\sigma}} \rangle_{L^2(\Gamma^{\sigma})} \quad (\text{A.5})$$

Using again the decomposition (A.3) results in:

$$\forall \mathbf{e} \in \mathcal{S} : a(\mathbf{e}, \mathbf{f}) - \langle \gamma^{\dagger}\{a\}(\mathbf{f}), \bar{\mathbf{e}}|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} = b(\mathbf{e}) - \langle \gamma\{b\}, \bar{\mathbf{e}}|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} \quad (\text{A.6})$$

This formula will be used in two particular situations:

---

### Boundary residual for an adjoint equation

Let  $a : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$  and  $b : \mathcal{S} \rightarrow \mathbb{R}$  be bilinear and linear form that can be decomposed as (A.3). If  $\mathbf{y} \in \mathcal{S}_0^2$  is a solution of the equation  $\forall \mathbf{s} \in \mathcal{S}_0 : a(\mathbf{s}, \mathbf{y}) = b(\mathbf{s})$  and  $\mathbf{e} \in \mathcal{S}$  satisfies  $a(\mathbf{e}, \mathbf{y}) = 0$ , then:

$$b(\mathbf{e}) = \langle \gamma\{b\} - \gamma^\dagger\{a\}(\mathbf{y}), \bar{\mathbf{e}}|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} \quad (\text{A.7})$$

### Boundary residual for an adjoint eigenproblem

Let  $a : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$  be a bilinear form that can be decomposed as (A.3b). If  $\mathbf{q} \in \mathcal{S}_0$  is a solution of the equation  $\forall \mathbf{s} \in \mathcal{S}_0 : a(\mathbf{s}, \mathbf{q}) = 0$  and  $\mathbf{e} \in \mathcal{S}$ , then:

$$a(\mathbf{e}, \mathbf{q}) = \langle \gamma^\dagger\{a\}(\mathbf{q}), \bar{\mathbf{e}}|_{\Gamma^d} \rangle_{L^2(\Gamma^d)} \quad (\text{A.8})$$

## A.2 Green's identities

This section is a reminder of some Green's identities useful to derive the strong form of the adjoint equations or to get the expression of the operator  $\gamma$  and  $\gamma^\dagger$  introduced in the section above.

Consider two tensor fields  $\mathcal{T}$  and  $\mathcal{S}$  of order  $m$  and  $m-1$  respectively. If these tensor fields are regular enough the following Green's identity is satisfied:

$$\int_{\Omega} \operatorname{div} \mathcal{T} : \mathcal{S} = - \int_{\Omega} \mathcal{T} : \nabla \mathcal{S} + \int_{\Gamma} \mathcal{T} : \mathcal{S} \otimes \mathbf{n}. \quad (\text{A.9})$$

Let us apply this general identity to some particular cases. Consider  $z$  a scalar field and  $\mathbf{a}, \mathbf{b}$  two vector fields of sufficient regularity. The three following Green's identities are helpful to handle the pressure term, the diffusion term and the convective term

## A. Mathematical tools

---

respectively:

$$\int_{\Omega} z \operatorname{div} \mathbf{a} = \int_{\Omega} \mathbf{a} \cdot \nabla z + \int_{\Gamma} z \mathbf{a} \cdot \mathbf{n} \quad (\text{A.10a})$$

$$-\int_{\Omega} \Delta \mathbf{a} \cdot \mathbf{b} + \int_{\Gamma} \nabla \mathbf{a} : \mathbf{b} \otimes \mathbf{n} = -\int_{\Omega} \nabla \mathbf{a} : \nabla \mathbf{b} = -\int_{\Omega} \mathbf{a} \cdot \Delta \mathbf{b} + \nabla \mathbf{b} : \mathbf{a} \otimes \mathbf{n} \quad (\text{A.10b})$$

$$\begin{aligned} \int_{\Omega} \nabla \mathbf{a} : \mathbf{c} \otimes \mathbf{b} &= \int_{\Omega} (\nabla \mathbf{a})^* : \mathbf{b} \otimes \mathbf{c} \\ &= -\int_{\Omega} \mathbf{a} \cdot \operatorname{div}(\mathbf{c} \otimes \mathbf{b}) + \int_{\Gamma} (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{n}) \\ &= -\int_{\Omega} \nabla \mathbf{c} : \mathbf{a} \otimes \mathbf{b} + (\mathbf{a} \cdot \mathbf{c}) \operatorname{div} \mathbf{b} + \int_{\Gamma} (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{n}) \end{aligned} \quad (\text{A.10c})$$

# Appendix B

## On the Navier–Stokes equations

### B.1 Geometry and notation

Consider a bounded open domain  $\Omega \subset \mathbb{R}^n$  and a partition of its boundary  $\Gamma = \Gamma^\sigma \cup \Gamma^d$ :  $\Gamma^d$  is the part of the boundary where the value of the velocity is prescribed (Dirichlet boundary condition) while  $\Gamma^\sigma$  is the part of the boundary where a flux is prescribed (Neumann or Robin boundary condition). Consider a different partition of the boundary  $\Gamma = \Gamma_- \cup \Gamma_+$ : given a velocity field  $\mathbf{u}$ , define the inflow boundary  $\Gamma_- = \{\mathbf{x} \in \Gamma \mid \mathbf{u} \cdot \mathbf{n} < 0\}$  and the outflow boundary  $\Gamma_+ = \{\mathbf{x} \in \Gamma \mid \mathbf{u} \cdot \mathbf{n} \geq 0\}$ . Finally, defined the intersection  $\Gamma_+^\sigma = \Gamma^\sigma \cap \Gamma_+$  and  $\Gamma_-^\sigma = \Gamma^\sigma \cap \Gamma_-$ .

### B.2 Boundary condition

The natural boundary condition is to prescribe the momentum flux on the inflow and the traction on the outflow:

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma^d \quad (\text{B.1a})$$

$$(-\mathbf{u} \otimes \mathbf{u} - p\mathbf{J} + 2\text{Re}^{-1}\overline{\nabla}\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}_- \quad \text{on } \Gamma_-^\sigma \quad (\text{B.1b})$$

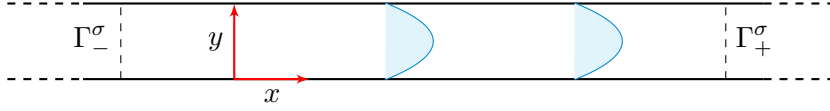
$$(-p\mathbf{J} + 2\text{Re}^{-1}\overline{\nabla}\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}_+ \quad \text{on } \Gamma_+^\sigma \quad (\text{B.1c})$$

However, this natural boundary condition is not always appropriate. On some occasions, the boundary does not have any physical significance and is used to crop a large or even infinite domain. Consider as an example the numerical study of an infinite channel Poiseuille flow depicted below.

The velocity inside the channel is  $\mathbf{u} = 4y(1 - y)\mathbf{e}_x$  and the pressure  $p = p_0 -$

## B. On the Navier–Stokes equations

---



$8\text{Re}^{-1}x$ . Therefore the traction on a vertical plane is  $(-p\mathbf{J} + 2\text{Re}^{-1}\overline{\nabla}\mathbf{u}) \cdot \mathbf{e}_x = -p_0\mathbf{e}_x + 2\alpha\text{Re}^{-1}(x\mathbf{e}_x - y\mathbf{e}_y)$ . The prescription of a zero traction at the outflow is certainly not the appropriate boundary condition. This is logical as the boundary  $\Gamma_+^\sigma$  is not a free surface; the wall after the boundary still constrains the flow. For such configurations, it is more suited to use the following boundary condition often called advective boundary condition:

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma^d \quad (\text{B.2a})$$

$$(-\mathbf{u} \otimes \mathbf{u} - p\mathbf{J} + \text{Re}^{-1}\nabla\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}_- \quad \text{on } \Gamma_-^\sigma \quad (\text{B.2b})$$

$$(-p\mathbf{J} + \text{Re}^{-1}\nabla\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}_+ \quad \text{on } \Gamma_+^\sigma \quad (\text{B.2c})$$

### B.3 Conservative form

The incompressible Navier-Stokes equations can be written in several equivalent forms that leads to slightly different discretized forms. The form that arises from the physics is called the strong conservative form.

#### Strong conservative form

Find  $(\mathbf{u}, p) \in \mathcal{S}^2$  that respect the condition (B.1) [or (B.2)] and the following equations on  $\Omega$ :

$$\begin{aligned} -\text{div}(-\mathbf{u} \otimes \mathbf{u} - p\mathbf{J} + 2\text{Re}^{-1}\overline{\nabla}\mathbf{u}) &= \mathbf{f} \\ \text{div } \mathbf{u} &= 0. \end{aligned} \quad (\text{B.3})$$

The weak form of this equation can be obtained using the Green's identity (A.9):

---

**Weak conservative form**

Find  $(\mathbf{u}, p) \in \mathcal{S}[\mathbf{g}]$  such that  $\forall(\mathbf{v}, q) \in \mathcal{S}$  :

$$\int_{\Omega} \left( -\mathbf{u} \otimes \mathbf{u} - p\mathcal{J} + \frac{\overline{\nabla \mathbf{u}}}{\text{Re}} \right) : \nabla \mathbf{v} + q \operatorname{div} \mathbf{u} + \int_{\Gamma^\sigma} \mathcal{G} : \mathbf{n} \otimes \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma^\sigma} \mathbf{h} \cdot \mathbf{v}$$
$$\text{with } \mathcal{G} = \begin{cases} \mathbb{1}_{\Gamma_+^\sigma}(\mathbf{u} \otimes \mathbf{u}) & \text{if (B.1) is used} \\ \mathbb{1}_{\Gamma_+^\sigma}(\mathbf{u} \otimes \mathbf{u}) - \text{Re}^{-1} \nabla \mathbf{u} & \text{if (B.2) is used} \end{cases} .$$
(B.4)

## B.4 Advective form

If we use the incompressibility relation  $\operatorname{div} \mathbf{u} = 0$  to simplify the term  $\operatorname{div}(\overline{\nabla \mathbf{u}})$ , we get a different form of the Navier–Stokes equations called strong advective form:

**Strong advective form**

Find  $(\mathbf{u}, p) \in \mathcal{S}^2$  that respect the condition (B.1) [or (B.2)] and the following equations on  $\Omega$ :

$$\begin{aligned} \nabla \mathbf{u} \cdot \mathbf{u} + \nabla p - \text{Re}^{-1} \Delta \mathbf{u} &= \mathbf{f} \\ \operatorname{div} \mathbf{u} &= 0. \end{aligned}$$
(B.5)

This form is widely used in numerical application, partly due to an easier implementation with finite differences codes.

The weak form of this equation can be obtained using the Green's identities (A.10):

**Weak advective form**

Find  $(\mathbf{u}, p) \in \mathcal{S}[\mathbf{g}]$  such that  $\forall(\mathbf{v}, q) \in \mathcal{S}_0$ :

$$\int_{\Omega} \nabla \mathbf{u} : \left( \mathbf{v} \otimes \mathbf{u} + \frac{\nabla \mathbf{v}}{\text{Re}} \right) - p \operatorname{div} \mathbf{v} + q \operatorname{div} \mathbf{u} - \int_{\Gamma^\sigma} \mathcal{G} : \mathbf{n} \otimes \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma^\sigma} \mathbf{h} \cdot \mathbf{v}$$
$$\text{with } \mathcal{G} = \begin{cases} \mathbb{1}_{\Gamma_-^\sigma}(\mathbf{u} \otimes \mathbf{u}) - \text{Re}^{-1} \nabla \mathbf{u} & \text{if (B.1) is used} \\ \mathbb{1}_{\Gamma_-^\sigma}(\mathbf{u} \otimes \mathbf{u}) & \text{if (B.2) is used} \end{cases} .$$
(B.6)

## B. On the Navier–Stokes equations

---

With the weak conservative form, the boundary term  $\mathcal{G}$  is simpler when the natural boundary condition is used. On the contrary, with the advective form, the term  $\mathcal{G}$  is simpler when the advective boundary condition is used. In this sense, the advective boundary condition (B.2) is the standard boundary condition<sup>1</sup> associated with the weak advective form (B.6) and the natural boundary condition (B.1) is the standard boundary condition associated with the weak conservative form (B.4).

---

<sup>1</sup>Sometimes called "no" boundary condition.

# Bibliography

- [1] ALLAIRE, G., JOUVE, F. & TOADER, A.M. (2002). A level-set method for shape optimization. *Comptes Rendus de l'Académie des Sciences*, **334**, 1125–1130. 30
- [2] BISMUT, J.M. (1973). Conjugate convex functions in optimal stochastic control. *Journal of Mathematical Analysis and Applications*, 384–404. 31
- [3] DAMBRINE, M. & KATEB, D. (2011). On the shape sensitivity of the first Dirichlet eigenvalue for two-phase problems. *Applied Mathematics and Optimization*, **63**, 45–74. 17
- [4] DE BOOR, C. (1978). *A practical guide to splines*. Springer. 5
- [5] EPPLER, K., SCHMIDT, S., SCHULZ, V. & ILIC, C. (2009). Preconditioning the pressure tracking in fluid dynamics by shape Hessian information. *Journal of Optimization Theory and Applications*, **141**, 513–531. 29
- [6] EVANS, L.C. (1998). *Partial differential equations*. American Mathematical Society. 33
- [7] GHANEM, R. & SPANOS, P.D. (2003). *Stochastic finite elements: a spectral approach*. Dover Publications. 30
- [8] GRAHAM, C. & TALAY, D. (2011). *Simulation stochastique et méthodes de Monte-Carlo*. Éditions de l'École Polytechnique. 30
- [9] GRINFELD, P. (2010). Hadamards formula inside and out. *Journal of Optimization Theory and Applications*, **146**, 654–690. 17
- [10] HADAMARD, J. (1908). *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, vol. 33. Imprimerie nationale. 17
- [11] HENNINGSON, D.S. (1996). Comment on "Transition in shear flows. Nonlinear normality versus non-normal linearity". *Physics of Fluids*, **8**, 2257. 32

## BIBLIOGRAPHY

---

- [12] HENROT, A. & PIERRE, M. (2005). *Variation et optimisation de formes: une analyse géométrique*, vol. 48. Springer. 13, 17
- [13] HEUVELINE, V. & STRAUSS, F. (2007). Hydrodynamic stability control in CFD by shape optimization. *PAMM*, **7**, 4140007–4140008. 17
- [14] JAMESON, A. (1988). Aerodynamic design via control theory. *Journal of Scientific Computing*, **3**, 233–260. 9
- [15] JAMESON, A. (1999). Re-engineering the design process through computation. *Journal of Aircraft*, **36**, 36–50. 9
- [16] JUNIPER, M.P. (2010). Triggering in the horizontal Rijke tube: non-normality, transient growth and bypass transition. *Journal of Fluid Mechanics*, **667**, 272–308. 30
- [17] KAROUI, N.E., PENG, S. & QUENEZ, M.C. (1997). Backward stochastic differential equations in finance. *Mathematical finance*, **7**, 1–71. 31
- [18] KIM, N.H., WANG, H. & QUEIPO, N.V. (2006). Efficient shape optimization under uncertainty using polynomial chaos expansions and local sensitivities. *AIAA journal*, **44**. 30
- [19] LERAY, J. (1934). Sur le mouvement d’un liquide visqueux emplissant l’espace. *Acta Mathematica*, **63**, 193–248. 3
- [20] MA, Z., AHUJA, S. & ROWLEY, C.W. (2010). Reduced-order models for control of fluids using the eigensystem realization algorithm. *Theoretical and Computational Fluid Dynamics*, **25**, 233–247. 30
- [21] MARGOLIN, L.G., RIDER, W.J. & GRINSTEIN, F.F. (2006). Modeling turbulent flow with implicit LES. *Journal of Turbulence*, **7**, N15. 31
- [22] MARSDEN, A.L., WANG, M., DENNIS, J.E. & MOIN, P. (2007). Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, **572**, 13. 9, 31
- [23] NADARAJAH, S.K., JAMESON, A. & ALONSO, J.J. (2002). Sonic boom reduction using an adjoint method for wing-body configurations in supersonic flow. *AIAA Paper*, **5547**. 9

- [24] PIRONNEAU, O. (1973). On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, **59**, 117–128. 9
- [25] PIRONNEAU, O. (1974). On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, **64**, 97–110. 9
- [26] SCHILLINGS, C., SCHMIDT, S. & SCHULZ, V. (2011). Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids*, **46**, 78–87. 30
- [27] SCHMID, P.J. & HENNINGSON, D.S. (2001). *Stability and transition in shear flows*, vol. 142. Springer Verlag. 29
- [28] SCHMIDT, S. & SCHULZ, V. (2010). Shape derivatives for general objective functions and the incompressible NavierStokes equations. *Control and Cybernetics*, **39**, 677–713. 15
- [29] SCHMIDT, S., ILIC, C., SCHULZ, V. & GAUGER, N.R. (2011). Airfoil design for compressible inviscid flow based on shape calculus. *Optimization and Engineering*, **12**, 349–369. 9
- [30] SIPP, D., MARQUET, O., MELIGA, P. & BARBAGALLO, A. (2010). Dynamics and control of global instabilities in open-flows: a linearized approach. *Applied Mechanics Reviews*, **63**. 3
- [31] SOKOLOWSKI, J. & ZOCHOWSKI, A. (1999). On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, **37**, 1251–1272. 30
- [32] STÜCK, A. & RUNG, T. (2011). Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, **47**, 22–32. 31
- [33] ZHOU, K., DOYLE, J.C. & GLOVER, K. (1996). *Robust and optimal control*. 30