



## POLITECNICO DI MILANO

---

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE  
Corso di Laurea Magistrale in Ingegneria Aeronautica

TESI DI LAUREA MAGISTRALE

### A robust open-source adjoint optimization method for external aerodynamics

Relatore

**Prof. Maurizio QUADRO**

Candidato

**Giuseppe SORGIOVANNI**

Matr. 838116

Correlatore

**Ing. PhD Raffaele PONZINI**

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Approaches to adjoint method . . . . .	2
1.2	Adjoint in CFD codes . . . . .	5
<b>2</b>	<b>Continuous adjoint</b>	<b>7</b>
2.1	Lagrangian-Based Approach (LBA) . . . . .	7
2.2	Lagrangian-Free Approach (LFA) . . . . .	11
<b>3</b>	<b>Force optimization</b>	<b>14</b>
3.1	Problem definition . . . . .	14
3.2	Pressure Force Formulation (PFF) . . . . .	16
3.3	Lagrangian-based Total Force Formulation ( $TFF_{LBA}$ ) . . . . .	20
3.4	Momentum Balance Formulation (MBF) . . . . .	21
3.5	Lagrangian-free Total Force Formulation ( $TFF_{LFA}$ ) . . . . .	24
3.6	Implementation in OpenFOAM . . . . .	28
3.6.1	Solvers . . . . .	28
3.6.2	Porting from v3.0 to v4.1 . . . . .	29
3.6.3	Solver's features . . . . .	29
<b>4</b>	<b>Study on 2D flows</b>	<b>31</b>
4.1	Computational mesh . . . . .	31
4.1.1	snappyHexMesh . . . . .	32
4.1.2	blockMesh . . . . .	33
4.1.3	SmoothCMeshGenerator (SCMG) . . . . .	35
4.2	Sensitivity dependence on mesh quality . . . . .	39
4.3	Sensitivity dependence on numerical schemes . . . . .	39
4.4	Test case A: Cylinder . . . . .	41
4.5	Test case B: Airfoil . . . . .	47
4.6	Discussion of methods . . . . .	52

4.6.1	Pressure Force Formulation . . . . .	52
4.6.2	Momentum Balance Formulation . . . . .	53
4.6.3	Total Force Formulation - LFA . . . . .	54
<b>5</b>	<b>Validation</b>	<b>55</b>
5.1	Parametrization . . . . .	55
5.2	Case setup . . . . .	56
5.3	Results . . . . .	57
<b>6</b>	<b>Conclusions</b>	<b>60</b>

# List of Figures

3.1	Problem layout for external aerodynamics . . . . .	15
4.1	NACA 0012 mesh generated with <i>snappyHexMesh</i> - far view . . . . .	33
4.2	Original <i>meshgen</i> vs modified: down-scaled 8K cells meshes. The same number of points on airfoil is kept for comparison . . . . .	34
4.3	Vertices layout on outer contour: example . . . . .	36
4.4	Vertices layout on inner domain: sampling every 10 lines . . . . .	37
4.5	Mesh for NACA 0012 at 2.5 AoA generated with SCMG - far view .	38
4.6	Mesh for NACA 0012 at 2.5 AoA generated with SCMG - near view	38
4.7	Comparison of meshes - Case B drag sensitivity computed with TFF <sub>LFA</sub> . . . . .	39
4.8	Comparison of numerical schemes - Case B drag sensitivity computed with TFF <sub>LFA</sub> . . . . .	40
4.9	Case A - Primal fields . . . . .	42
4.10	Case A - Pressure Force Formulation: adjoint fields . . . . .	43
4.11	Case A - Momentum Balance Formulation: adjoint fields . . . . .	44
4.12	Case A - Total Force Formulation: adjoint fields . . . . .	45
4.13	Case A - Drag sensitivity comparison (green and red curves overlap)	46
4.14	Case B - Primal fields . . . . .	48
4.15	Case B - Pressure Force Formulation: adjoint fields . . . . .	49
4.16	Case B - Momentum Balance Formulation: adjoint fields . . . . .	50
4.17	Case B - Total Force Formulation: adjoint fields . . . . .	51
4.18	Case B - Drag sensitivity comparison . . . . .	52
4.19	Case B - Drag sensitivity comparison: magnified view . . . . .	53
5.1	Set of sine-squared shape functions . . . . .	56
5.2	$5 \cdot 10^{-3}$ peak displacement in nodes 36, 50, 71 . . . . .	56
5.3	Caption . . . . .	56

5.4	Finite differences vs adjoint methods: $C_D$ gradient . . . . .	58
5.5	Finite differences vs adjoint methods: $C_L$ gradient . . . . .	58
5.6	Finite differences vs Total Force Formulation - LFA: $C_D$ gradient . .	59
5.7	Finite differences vs Total Force Formulation - LFA: $C_L$ gradient . .	59

# Abstract

Adjoint method is an hot topic in computational fluid dynamics, as it represents the most viable option to apply gradient-based optimization to problems that require the solution of flow equations. The main goals of the present work are the enhancement of available methodologies and the search for new and more performing ones, that can achieve high accuracy in a finite volume implementation. Two different approaches to formulate continuous adjoint problem for aerodynamic optimization are explored. Boundary conditions and surface sensitivity are derived for a number of suitable cost function definitions, which are then discussed and implemented in a Open Source framework. The well known issue of higher order derivatives in analytic expressions of sensitivity is successfully handled in two of the variants: the first being completely new and the second adapted from the work of Castro et al. [1]. Being accuracy the main weakness of continuous adjoint method, a validation of sensitivity gradients based on finite differences is performed. The collected results emphasize the importance of having looser discretization requirements on gradients' evaluation, as the best results are achieved by the two aforementioned formulations. The last one, in particular, succeeded in providing a very accurate evaluation of both drag and lift sensitivities.

**Keywords:** Continuous adjoint, Aerodynamic optimization, Formulations, Open-FOAM, Open Source

# Sommario

Il metodo aggiunto è un argomento di grande attualità nell’ambito della CFD, in quanto costituisce la migliore alternativa per l’utilizzo di schemi gradient-based in problemi di ottimizzazione aerodinamica. Questa metodologia consente di calcolare in maniera estremamente efficiente il gradiente di un determinato funzionale di costo, dipendente dalle variabili di flusso, rispetto a un set di variabili di design (o controllo), la cui variazione ha effetto sulla soluzione del problema fluidodinamico. Nello specifico il costo computazionale richiesto dal calcolo del gradiente per una singola funzione obiettivo è solitamente di poco superiore a quello di una soluzione del problema diretto, ovvero delle equazioni di governo. L’alternativa, costituita dalle differenze finite, richiede invece un numero di valutazioni proporzionale alla dimensione del vettore di design. Poiché il costo unitario di una simulazione di CFD è elevato e l’ottimizzazione di forma richiede usualmente un numero elevatissimo di parametri, la prima opzione risulta l’unica percorribile, da qui l’interesse per il metodo aggiunto.

Nel continuo il controllo viene definito come un campo. Nel caso dell’ottimizzazione topologia questo è costituito da una distribuzione volumetrica di impedenza, ottenuta aggiungendo alle equazioni il termine di Darcy mentre, nel caso dell’ottimizzazione di forma, da un campo di spostamenti superficiali. Quest’ultima impostazione è quella adottata, essendo particolarmente adatta ai flussi non confinati.

Il presente lavoro ha potuto contare sul bagaglio conoscitivo costruito da un cospicuo numero di tesi pregresse, in particolare dai lavori di R. Pieri [2], M. Murari [3] e L. Palma [4]. Gli obiettivi prefissati sono il miglioramento delle metodologie già disponibili, come il solver *adjointOptFoam*, sviluppato e testato nelle ultime due tesi, e la ricerca di nuove formulazioni che possano garantire risultati accurati nell’ambito dei volumi finiti. In continuità con la tradizione dei precedenti lavori, l’implementazione dei solver e dei numerosi script sviluppati appositamente per il post-processing, la visualizzazione dei risultati e la sottomissione dei job è stata

---

condotta in ambienti Open Source.

Due diversi approcci nella formulazione dell’aggiunto continuo sono stati tratti dalla letteratura: il primo impiega il formalismo della *Lagrangiana* [5], il secondo rientra nell’approccio classico del controllo distribuito [1]. Le condizioni al contorno e la derivata di sensitività (trattasi di una derivata funzionale), sono state ricavate per una serie di funzioni di costo basate sulla risultante delle azioni aerodinamiche. Si sono così ottenute quattro diverse varianti. Il problema principale che inficia l’accuratezza dei metodi è la presenza di derivate di ordine superiore nell’espressione analitica della sensitività. Questi termini presentano lo stesso ordine delle equazioni di governo e sono perciò approssimati con un’accuratezza inferiore rispetto alle variabili di flusso; ciò comporta un’errore importante nel calcolo della sensitività. Questo problema, ben noto in letteratura, è stato risolto in due delle nuove varianti: la prima rappresenta uno sviluppo completamente nuovo, basato su una definizione alternativa della funzione di costo, mentre la seconda è stata ottenuta riproducendo stessa metodologia usata in Castro et al. [1] nell’ambito dei flussi comprimibili.

Tre delle quattro varianti sono state implementate come solver *OpenFOAM* separati, ognuno dotato del proprio set specifico di BC. Per l’implementazione si è mantenuta l’impostazione di base del solver *adjointOptFoam*, del quale è stato fatto il porting dalla versione 3.0 alla 4.1 di *OpenFOAM*.

Trattandosi di uno studio centrato sui metodi si è scelto di restringere la campagna di prove a geometrie 2D e flussi laminari. Il primo aspetto è ovviamente orientato alla riduzione della complessità e dell’onere computazionale, mentre il secondo permette di evitare l’errore numerico introdotto dagli schemi del prim’ordine tipicamente usati per la discretizzazione dei modelli di turbolenza. I due casi test principali, qui riportati, sono il flusso stazionario intorno a un cilindro e quello attorno ad un profilo aerodinamico a bassa incidenza.

Poiché, come è noto, la soluzione aggiunta è molto sensibile alla risoluzione e alla regolarità della griglia computazionale, si è prestata particolare attenzione a questo aspetto. Sono state sperimentate sia mesh strutturate che non, alla fine optando per le prime, che rappresentano la miglior scelta nel caso di geometrie 2D semplici. Successivamente diversi layout e settaggi sono stati esplorati all’interno dell’applicazione *blockMesh* di *OpenFOAM*. Infine, con lo scopo di ottenere un miglior controllo su raffinamento e regolarità, si è implementato ex novo un generatore di mesh strutturate usando il linguaggio e le librerie *Octave*. Il software si basa sull’impiego di *spline cubiche hermitiane* per la parametrizzazione della geometria

in input e di *curve di Bezier* per il controllo della disposizione dei vertici. Il design della mesh del profilo si è ispirato al noto set di griglie sviluppato dalla NASA per il NACA 0012 [6].

Infine, essendo l'accuratezza delle derivate di sensitività il punto debole delle formulazioni continue, soggette al problema dei gradienti inconsistenti, si è deciso di procedere alla validazione tramite confronto con le differenze finite. La procedura è stata applicata al caso del profilo, rivelatosi più critico durante le prove. Una parametrizzazione deformativa con funzioni di forma a supporto compatto è stata introdotta ex novo allo scopo di campionare efficacemente la sensitività nodale, oltre che di ridurre il numero di parametri e di applicare deformazioni regolari.

Il confronto con il gradiente esatto ha dimostrato come le due nuove formulazioni siano capaci di risultati molto più accurati. In particolare, mentre il gradiente del  $C_L$  è calcolato in maniera sufficientemente precisa da tutti e tre i solver, quello del  $C_D$  risulta molto sovrastimato dalla formulazione originale. La somiglianza qualitativa con il profilo di sensitività esatto suggerisce come il metodo in questione possa indicare una strategia corretta, seppur sub-ottimale, per la minimizzazione della resistenza aerodinamica. La sorgente dell'errore è da ricercare in una discrettizzazione non sufficientemente accurata per la valutazione della sensitività. Infine la formulazione “alla Castro” ha prodotto dei gradienti estremamente fedeli alle differenze finite, cioè, unito alla maggior flessibilità che la caratterizza, ne decreta il primato sulle altre implementazioni.

**Parole chiave:** Problema aggiunto continuo, Ottimizzazione aerodinamica, Formulazioni, OpenFOAM, Open Source

# Chapter 1

## Introduction

Adjoint method is a very popular topic in fluid dynamic optimization, the reason is simple: it provides a wealth of information at a cost barely higher than a common CFD solution. The information comes in the shape of sensitivity maps, surface (or volume) distributions that quantify the response to a surface displacement (or cell occlusion) w.r.t. a given merit function or parameter. This data can be used to unveil the most sensitive portions of a design and serve as a design aid tool or it can be recast to provide the gradients w.r.t. a geometric parametrization and then used within an optimization cycle. Advantages come not without drawbacks, as J. Peter and R. Dwight [7] remark on the “availability of cheap, accurate gradients”:

*A good example is the lack of reliable adjoint solvers for Navier-Stokes problems until recently. Initially the difficult and limited adjoint formulation in the continuous context slowed development. Later the stability of iterations for the viscous adjoint in many situations proved problematic. Even today there are few adjoint solvers that apply linearized turbulence models to large problems – predominantly due to poor stability – instead they rely on constant eddy-viscosity approximations, resulting in an associated gradient error.*

Although the topic is dealt with in many papers and essays, it's hard to find a thorough discussion. So, reproducing the results showcased in literature is not an easy task because of the lack of relevant details.

During the initial phase, some of the most important sources [5] [8] [1] [9] [10] [11], different in the theoretical approach and discussed topics, were selected in the literature. Following an in-depth analysis, two general and organic methodologies have been drawn up, which can be applied to the case of our interest, i.e., aerodynamic performance optimization. The present work is in continuity with

the achievements and experiences gained by previous master's theses on the topic, with special reference to the work of R. Pieri [2], M. Murari [3] and L. Palma [4]. Our main goals are: the improvement of available methods, such as *adjointOptFoam* solver, developed and tested in the last two theses, and the exploration of new formulations capable to achieve high levels of accuracy in a finite volumes implementation. Following the tradition of previous works, the CFD code, as well as any other script used in this study, have been developed using open source packages, namely: *OpenFOAM* and *Octave*. The usage of *OpenFOAM* libraries allows for straightforward modification of existing code and the desired freedom in the choice of numerical methods. In addition, the finite volumes method, which represents the current standard in industrial applications, fulfills the requirements of usability and scalability that are needed to deal with big scale problems.

The study deals with steady incompressible NS, to which we'll refer in the following. The scope of work is restricted at present to laminar flows, with underlying thought of providing a stable foundation for further developments.

## 1.1 Approaches to adjoint method

A number of approaches and model assumptions are possible in dealing with adjoint method, a brief summary of the most relevant is presented. The first choice is between continuous and discrete adjoint formulations. The former is the “optimize then discretize” approach, which consist in the derivation of the adjoint of primal equations, which is only then discretized and solved. The latter is the “discretize then optimize” approach, which conversely uses the adjoint of the algebraic system stemming from primal equation discretization.

A typical issue of continuous adjoint is inconsistent gradients. As one can imagine, solving a different PDE problem which is the theoretical adjoint of the primal one doesn't grant the sensitivity to be consistent with the actual response of the direct problem, which is itself approximated. The fact adjoint equations depends on the primal solution and the final product, i.e. the sensitivity map, is obtained post-processing both primal and adjoint fields, explains why error is so easily amplified. (At present error estimation results are lacking). The discrete counterpart is instead consistent by definition, yet if strictly speaking it provides the response of an algebraic system to the perturbation of a set of its coefficients. The reason why continuous approach is attractive lies in the fact it's not discretization dependent, and hence much less implementation effort is required, or from a different

point of view, the same formulation holds for many problems. In most cases only BCs and/or the sensitivity expression need to be changed if a different primal case layout or merit function is considered. This also uncouples to a reasonable extent primal and adjoint discretizations: not only different schemes, but even different grids can be employed. This freedom is indeed a strength and a weakness as the inner consistency of results can be, more or less willingly, altered. An interesting development is reported in [12] in the context of external car aerodynamics: recurring to an approximation involving a turbulence model, the adjoint of the time averaged LES/DES equations has been derived leading to a result which is formally identical to the adjoint RANS equations. The flow field obtained averaging a DES solution has been fed to the adjoint equations providing a primal solution that is richer in information and resulting in a sensitivity map which is more realistic. A mixed approach like this is solely possible in the continuous case, as it's built upon a primal problem that is not meant to be solved at all. Another advantage of continuous approach is the availability of the fields of adjoint variables, whose inspection can help giving some 'physical' interpretation of adjoint solution or pointing out issues and opportunities. An interesting development is for instance "adjoint wall functions" in [13]

An issue common to both approaches is stability. The inspection of continuous adjoint RANS equations shows that we deal with a convection-diffusion PDE problem with the transport field being the opposite of flow velocity. The problem is linear (as the derivation deals with first variations), and this means the superposition principle applies. A particular feature is the symmetric convective stress tensor, that can be split in the usual convection term plus the so called "adjoint transpose convection" term (briefly referred to as ATC). Both this term and the transport (dominated) nature of the problem are the roots of instability issues. The continuous adjoint approach has been pursued in this work as it offers the desired flexibility and it's suitable to be implemented as an OpenFOAM solver.

Another aspect is the way control (or design) variables are defined, two different setup are commonly used: topological shape optimization and surface shape optimization. In the first setup the control acts through a Darcy term added to the equations so to impede the flow in unfavorable regions of the domain. The control is a volume distribution (of impedance), and so it is sensitivity. This setup is simple in principle and usually grants straightforward expression for sensitivity, moreover the same mesh can be used throughout the optimization process. More effort must be devoted to pilot optimization to a "binary" impedance distribution with the

meaning of either “flow” or “no flow”, as the aim of the whole process is the redefinition of the wall shape. The advantage of a fixed mesh comes with the downside of a limited resolution in the optimized surface. Adaptive mesh refinement could mitigate the issue. Control is not forced, unless constrained accordingly, to act in the vicinity of the surface or to keep the original topology of the solid domain, this is the reason for the name of the method. In the second setup the control is defined as a displacement distribution applied to the surface to be optimized, for this reason we deal with surface sensitivities. The expression of sensitivity can be less trivial concerning derivation and evaluation, moreover mesh deformation or remeshing is needed to evaluate the new designs. Being this process very demanding and delicate is not infrequent to perform it once, stopping to a sub-optimal design instead of performing an automated design loop.

As matter of fact topological optimization is mostly used in internal flows, while surface deformation is preferred in external aerodynamics or as an additional step to refine the “castellated” output of topological optimization. Making no exception, surface deformation is the choice for this work.

Finally also the definition of the objective function, which can be volume based, surface based or even mixed, constitutes another possible bifurcation. When the objective is a functional of volume fields additional volume terms make their appearance in equations, and it can be the case that the adjoint velocity is no more divergence free. In the case of surface functionals the influence of the chosen parameter is only on boundary conditions, and equations are kept unchanged: this means the objective function can be varied as easily as changing BCs. Force coefficients are with no doubt the most common choice of cost function for external aerodynamics and constitutes the focus of our investigation. The case in which both control and objective function are defined on the same boundary give raise to a more articulate expression of sensitivity, that needs special care. This issue has been addressed in two different ways. (See sections 3.4 and 3.5)

Another choice, which is possible in both approaches, is whether deriving or not the adjoint of the turbulence model. If the latter is neglected (frozen turbulence hypothesis) the effect is a sensitivity that doesn’t take into account the variation of eddy viscosity w.r.t control. This falls in the topic of inner consistency between primal and adjoint problems. If we think of an optimization cycle the issue is a poorer quality of gradients, possibly leading to a sub-optimal design. Even if the hypothesis has been widely used so far in order to avoid extra efforts in derivation and solution, the trend is clearly towards removing it.

## 1.2 Adjoint in CFD codes

While developments have been around for a couple decades by now, adjoint method is still far from being an established tool in free and even commercial CFD packages. On one side adjoint solvers can be found in some popular commercial packages often as additional modules, and with accuracy being not the main focus, on the other side they exist in a number of academic ad hoc implementations that are often superior on a numerical basis, but yet unsuitable for daily industrial application. A separate class is in-house code developed from companies in the aerospace sector and space agencies. A brief list of some of the available codes is given below.

**OpenFoam** OpenFOAM natively embeds a topological optimization solver named *adjointShapeOptimizationFoam*, yet the solver dates back to 2007 and no further developments or much documented usage are known to date. Detailed information can be found in [14]. There exist also a number of custom versions that are not open source, for example the solver developed by Engys, based on the work of E. de Villiers and C. Othmer (visit [15]) and featuring shape and topology optimization based on the continuous formulation. A discrete adjoint version of OpenFOAM is developed at STCE, RWTH Aachen university using algorithmic differentiation (AD). Only the AD software *dc* is currently available for download (for more details visit [16]).

**SU2** Remarkable open-source software featuring adjoint problem solution. Focused on compressible RANS equations and thought for aeronautical applications, it features both continuous and discrete adjoint as well as a number of additional modules ranging from non conventional fluid dynamics, to electrodynamics. Incompressible flow solution is implemented as a sub-case of the compressible solver. Theoretical foundations and details can be found in [9].

**ANSYS Fluent** An adjoint module, called *ANSYS adjoint solver* is available in the category of shape optimization tools. According to the available documentation, the solver is based on discrete adjoint and features internal and external aerodynamic optimization w.r.t. force components or pressure drop. A Bernstein polynomial-based morphing scheme is present and used to smooth sensitivity and apply deformations. A mesh deformation tool based on RBF (Radial-Basis-Functions) is advertised aside of the module.

**STAR CCM+** Adjoint solver is available from v8, it features analysis of ducted and external flows. The set of cost functions includes force and moment coefficients, pressure drop and flow uniformity. Sensitivities are available w.r.t. flow parameters, mesh coordinates and boundary conditions. The second ones can be used to have an insight on relevant portions of the mesh and improve mesh design. Morphing is based on parameters, such as control points, that allow for a restriction of design space and filter sensitivities.

# Chapter 2

## Continuous adjoint

Two alternative approaches have been considered, the first is based on the Lagrangian formalism while the second falls back in the classical theory of optimal control for PDE. The difference in results is subtle in some cases while it appears to be relevant in the case of force optimization. In the following the two methodologies are presented in an abstract and general fashion, while they will be particularized in the subsequent chapters.

### 2.1 Lagrangian-Based Approach (LBA)

The main references for this section are [8] and [5]. The first covers steady incompressible NS and Euler equations and provides some useful results such as metrics-free gradient evaluation. The second, dealing with ducted flows, provides the basis for the extension to external aerodynamics, a topic which is dealt in lesser detail in [12]. In the following this approach will be referred to as LBA.

Let's define the objective function  $J$  and the control field  $\mathbf{b}$

$$J = \int_{\Gamma_J} g(\mathbf{U}, \mathbf{n}) \quad \text{with } \Gamma_J \equiv \Gamma_{body} \quad (2.1)$$

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{b}, \quad \mathbf{b} = \alpha_1 \mathbf{t}_1 + \alpha_2 \mathbf{t}_2 + \beta \mathbf{n} \quad \text{on } \Gamma_b \equiv \Gamma_{body} \quad (2.2)$$

wherein  $\mathbf{U}$  represents the field of flow variables,  $\mathbf{n}$  the boundary normals,  $\mathbf{t}_1$  and  $\mathbf{t}_2$  two arbitrary, non-collinear, boundary tangent versors and  $g$  the surface “density” of the objective function.  $\Gamma_J$  and  $\Gamma_b$  represent the boundary portions on which  $g$  and  $\mathbf{b}$  are respectively defined, they both identify with the body surface  $\Gamma_{body}$ . When we consider small perturbation of the body surface, control can be reduced

to a distribution of normal displacement

$$\delta \mathbf{x} = \delta \mathbf{b} = \delta \beta \mathbf{n} \quad (2.3)$$

It's the case of the information embed in sensitivity map, which is a first order derivative. For this reason in the following variations are expressed w.r.t.  $\beta$  instead of  $\mathbf{b}$ . Flow equations and BCs are represented in compact form

$$\mathbf{R}(\mathbf{U}) = \mathbf{0} \quad \text{on } \Omega(\mathbf{b}) \quad (2.4)$$

$$\mathbf{B}(\mathbf{U}) = \mathbf{0} \quad \text{on } \partial\Omega(\mathbf{b}) \quad (2.5)$$

The shape optimization problem can be expressed as follows

$$\text{minimize } J(\mathbf{U}, \mathbf{b}) \text{ subject to } \mathbf{R}(\mathbf{U}) = \mathbf{0} \quad (2.6)$$

Using the continuous analogous of I type lagrangian multipliers we can reformulate the constrained optimization problem into an unconstrained one.  $J$  is complemented with the constraint so to obtain the Lagrange function:

$$L(\mathbf{U}, \mathbf{b}, \Psi) := J + \int_{\Omega(\mathbf{b})} \Psi \cdot \mathbf{R} \quad (2.7)$$

wherein  $\Psi$  is the Lagrange multipliers field. (Comparing to the finite-dimensional case inner product between the vector of multipliers and the vector of constraints is substituted with  $L^2$  inner product between fields). It must be stressed that from this point on the flow variables  $\mathbf{U}$  are not required to be compliant with the flow equations as this is granted only upon minimization of  $L$ , this requires its total variation to vanish

$$\delta L = \delta_{\mathbf{U}} L + \delta_{\beta} L + \delta_{\Psi} L = 0 \quad (2.8)$$

If we evaluate this variation in a “feasible” point, i.e. with  $\mathbf{U}$  being a flow solution, it simplifies to

$$\delta L = \delta_{\mathbf{U}} L + \delta_{\beta} L \quad (2.9)$$

If moreover the multipliers are chosen such that the variation w.r.t. flow variables is zero, it further reduces to

$$\delta L = \delta_{\beta} L \quad (2.10)$$

Here we have an handy expression of the total variation of  $L$  from which the surface

map can be extracted. It's the case that the PDE problem that grants the correct choice of  $\Psi$  is the adjoint problem. Let's see it more in deep: the problem reads

$$\text{find } \Psi : \quad \delta_{\mathbf{U}} L(\delta \mathbf{U}, \Psi) = 0 \quad \forall \delta \mathbf{U} \quad (2.11)$$

The dependence on the linearization point  $(\mathbf{U}, \mathbf{b})$  is hidden for clarity. Let's recall the definition of the adjoint in functional analysis (see for instance chapter 2 of [17]). Taking the bilinear form  $a(u, v)$  with  $u, v$  belonging to a given Hilbert space  $V$ , its adjoint  $a^*$  is such that

$$a^*(u, v) = a(v, u) \quad \forall v, u \in V \quad (2.12)$$

Now if  $a(u, v)$  is the bilinear form typically found in the weak form of a linear PDE problem, with  $u$  representing unknowns and  $v$  test functions, the adjoint problem is obtained switching unknowns and test functions. This is exactly what we do in eq.(2.11) solving for  $\Psi$ . This is apparent expanding the first variation of  $L$

$$\delta_{\mathbf{U}} L(\delta \mathbf{U}, \Psi) = \delta_{\mathbf{U}} J(\delta \mathbf{U}) + \int_{\Omega} \Psi \cdot \delta_{\mathbf{U}} \mathbf{R}(\delta \mathbf{U}) \quad (2.13)$$

In the last integral we can recognize  $a(u, v)$  from the weak form of the the linearized flow equations, with  $\Psi$  being the test function and  $\delta \mathbf{U}$  the unknown. Solving instead for  $\Psi$  and having  $\delta \mathbf{U}$  as test function, the integral is indeed the adjoint of the linearized flow equations. In order to obtain adjoint equations in strong form and the related boundary conditions we need first to lower the order of the test function integrating by parts so to get

$$\int_{\Gamma_J} \delta_{\mathbf{U}} g(\delta \mathbf{U}) + \int_{\Omega} \delta \mathbf{U} \cdot \mathbf{A}^* + \int_{\partial\Omega} \mathbf{C}(\Psi, \delta \mathbf{U}) = 0 \quad \forall \delta \mathbf{U} \quad (2.14)$$

Adjoint equations are then obtained by the vanishing of the volume integral

$$\mathbf{A}^*(\Psi; \mathbf{U}) = \mathbf{0} \quad \text{on } \Omega(\mathbf{b}) \quad (2.15)$$

wherein the dependence on the linearization point  $(\mathbf{U}, \mathbf{b})$  is now shown.

Boundary conditions are derived from the vanishing of the boundary integrals

$$\int_{\Gamma_J(\mathbf{b})} \delta_{\mathbf{U}} g(\delta \mathbf{U}; \mathbf{U}, \mathbf{n}) + \int_{\partial\Omega(\mathbf{b})} \mathbf{C}(\Psi, \delta \mathbf{U}; \mathbf{U}) = 0 \quad \forall \delta \mathbf{U} \quad (2.16)$$

The next step is substituting the primal BCs.

$$\mathbf{U} : \quad \mathbf{B}(\mathbf{U}) = \mathbf{0} \quad (2.17)$$

At this point adjoint BCs are yet undetermined, this is overcome in [5] enforcing feasible  $\delta\mathbf{U}$ , i.e.

$$\delta\mathbf{U} : \mathbf{B}(\mathbf{U} + \delta\mathbf{U}) = \mathbf{0} \quad (2.18)$$

Adjoint BCs are now available in strong form

$$\mathbf{D}(\Psi; \mathbf{U}) = \mathbf{0} \quad \text{on } \partial\Omega(\mathbf{b}) \quad (2.19)$$

The adjoint problem is now fully defined by (2.15) and (2.19). As the current deformation  $\mathbf{b}$  is an unnecessary information in spot evaluation of sensitivity it has been dropped from notation. With the underlying meaning of linearization point equal to undeformed condition.

Finally the expression for sensitivity must be derived. Sensitivity is defined as the  $G$  term in

$$\delta_\beta L = \int_{\Gamma_J} G \delta\beta \quad (2.20)$$

This is referred to the case in which  $\mathbf{U}$  is the flow solution and  $\Psi$  the adjoint solution. Hence some manipulation to the following expression is needed

$$\delta_\beta L = \delta_\beta J + \delta_\beta \int_{\Omega} \Psi \cdot \mathbf{R} \quad (2.21)$$

The first term is the so called geometric term, it includes the direct dependence of  $J$  on the body deformation. This can be computed either analytically or by finite differences, yet the former is more consistent with a continuous approach. Expanding it

$$\delta_\beta J = \delta_\beta \int_{\Gamma_J} g(\mathbf{U}, \mathbf{n}) = \int_{\delta\Gamma_J} g(\mathbf{U}, \mathbf{n}) \quad (2.22)$$

This term is equal to the variation of the integral w.r.t. its integration domain. The computation of the second term on the right is not trivial as it requires evaluating how equation transform w.r.t. the domain deformation. This requires equation to be written in general coordinates. A metrics free approach is followed in [5] and due to [8]. If we consider admissible  $\delta\mathbf{U}$  the total variation of the flow equation will be zero

$$\delta\mathbf{R} = \delta_\beta \mathbf{R} + \delta_{\mathbf{U}} \mathbf{R} = \mathbf{0} \quad (2.23)$$

We can substitute  $\delta_\beta \mathbf{R}$  in eq.(2.21) and get

$$\delta_\beta L = \delta_\beta J - \int_{\Omega} \Psi \cdot \delta_{\mathbf{U}} \mathbf{R} \quad (2.24)$$

The volume integral has already been encountered and can be substituted with the second and third terms of (2.16). As the first of the two is zero when adjoint equations are fulfilled only boundary integrals remain

$$\delta_\beta L = \delta_\beta J(\delta\beta; \mathbf{U}) - \int_{\partial\Omega} \mathbf{C}(\delta\mathbf{U}; \boldsymbol{\Psi}, \mathbf{U}) \quad (2.25)$$

The last step is obtaining an expression depending on  $\delta\beta$  only. What is missing is the relation  $\delta\mathbf{U}(\delta\beta)$  (existing because by hypothesis  $\delta\mathbf{U}$  is feasible, making of  $\delta\beta$  the only free variation). An approximation of this relation is suggested by Soto and Löhner:

$$\delta\mathbf{U} \simeq \delta\beta(\mathbf{n} \cdot \nabla)\mathbf{U} \quad (2.26)$$

Finally sensitivity is obtained from the integrand of

$$\delta_\beta J(\delta\beta) - \int_{\partial\Omega} \mathbf{C}(\delta\beta(\mathbf{n} \cdot \nabla)\mathbf{U}) \quad (2.27)$$

## 2.2 Lagrangian-Free Approach (LFA)

The main reference for this section is [1], whose methodology has been abstracted and then ported from compressible to incompressible steady NS in section 3.5. For brevity we'll refer to this approach as LFA. Considering objective function and control defined as in eq.s (2.1-2.2), the total variation of  $J$  is

$$\delta J = \int_{\delta\Gamma_J} g(\mathbf{U}, \mathbf{n}) + \int_{\Gamma_J} \frac{\partial g}{\partial \mathbf{U}} \delta\mathbf{U} \quad (2.28)$$

where the partial derivative notation stands for Gateaux derivative (remark: it can be expressed this way if it's linear). The two integrals in the r.h.s. represent respectively the geometric term and the field term. An important difference with the previous derivation is that from the beginning  $\delta\mathbf{U}$  is dependent on  $\delta\beta$ , for this reason the total variation is indeed the variation w.r.t. control.

Governing equations are handled in conservation form and vector notation in the reference article, as this is typical in compressible flows literature.

$$\nabla \cdot F(\mathbf{U}) = \mathbf{0} \quad \text{on } \Omega(\mathbf{b}) \quad (2.29)$$

Here the same general form of eq. (2.4-2.5) is used instead.

The adjoint flow equations are derived explicitly as opposed to being the by-product of the stationary condition of a Lagrangian. The procedure requires first

linearization (let's consider  $\mathbf{U}$  the flow solution and  $\mathbf{b} = 0$ ), leading to

$$\delta \mathbf{R}(\delta \mathbf{U}) = \mathbf{0} \quad \text{on } \Omega \quad (2.30)$$

$$\delta \mathbf{B}(\delta \mathbf{U}) = \mathbf{0} \quad \text{on } \partial\Omega \quad (2.31)$$

Then the weak form of linearized equations

$$\int_{\Omega} \Psi \cdot \mathbf{A} \delta \mathbf{U} = 0 \quad \forall \Psi \quad (2.32)$$

with  $\mathbf{A}$  being the Gateaux derivative of  $\mathbf{R}(U)$ . At this point unknown and test function are switched and performing integration by parts adjoint weak form is derived

$$\int_{\Omega} \delta \mathbf{U} \cdot \mathbf{A}^* + \int_{\partial\Omega} \mathbf{C}(\Psi, \delta \mathbf{U}) = 0 \quad \forall \text{admissible } \delta \mathbf{U} \quad (2.33)$$

where admissible  $\delta \mathbf{U}$  comply with linearized flow equations (2.30-2.31) on a domain perturbed by arbitrary small deformations  $\delta \beta$  of  $\Gamma_b$ . Let's extract adjoint equations and keep BCs in weak form

$$\mathbf{A}^*(\Psi; \mathbf{U}) = \mathbf{0} \quad \text{on } \Omega, \quad \int_{\partial\Omega} \mathbf{C}(\Psi, \delta \mathbf{U}; \mathbf{U}) = 0 \quad \forall \text{admissible } \delta \mathbf{U} \quad (2.34)$$

It can be noted that adjoint equations are identical to the previously derived (2.15).

The next step involves the recognition of an objective function related term in the integral on the body portion of the boundary. Precisely for a given BC assignment of  $\Psi$  one can collect the second integral in eq. (2.28). (Detailed in section 3.5)

$$\int_{\Gamma_J} \frac{\partial g}{\partial \mathbf{U}} \delta \mathbf{U} \quad (2.35)$$

Collecting the remaining integrals in  $I_{eq}$  we get

$$\int_{\Gamma_J} \frac{\partial g}{\partial \mathbf{U}} \delta \mathbf{U} = -I_{eq}, \quad \text{if } \mathbf{D}_{\Gamma_{body}}(\Psi) = \mathbf{0} \quad (2.36)$$

gaining both the expression of the field term of sensitivity and the adjoint BCs on the body. The feasibility of this step restricts the present method to a family of admissible objective functions. This represents the main limit of the lagrangian-free approach. At this point let's substitute  $\delta \mathbf{U}$  in boundary integrals, expanding condition (2.31)

$$\delta \mathbf{U} : \frac{\partial \mathbf{B}}{\partial \mathbf{U}} \delta \mathbf{U} = \mathbf{0} \quad \text{on } \partial\Omega \setminus \Gamma_{body} \quad (2.37)$$

and considering  $\mathbf{U}$  depends on  $\delta\beta$  also through the deformation of  $\Gamma_{body}$ , i.e.  $\mathbf{U}(\mathbf{x}(\delta\beta))$ , its total variation reads  $\delta\mathbf{U} + \delta\mathbf{x} \cdot \nabla\mathbf{U}$ . Hence

$$\delta\mathbf{U} : \frac{\partial\mathbf{B}}{\partial\mathbf{U}}(\delta\mathbf{U} + \delta\beta\mathbf{n} \cdot \nabla\mathbf{U}) = \mathbf{0} \quad \text{on } \Gamma_{body} \quad (2.38)$$

This gives: the remaining BCs

$$\mathbf{D}_{\partial\Omega \setminus \Gamma_{body}}(\Psi) = \mathbf{0} \quad (2.39)$$

and term  $I_{eq}$  explicitly depending on control

$$-I_{eq} = \int_{\Gamma_J} G_2 \delta\beta \quad (2.40)$$

with reference to the full expression of sensitivity

$$\delta J = \int_{\Gamma_J} G_1 \delta\beta + \int_{\Gamma_J} G_2 \delta\beta \quad (2.41)$$

Finally the missing term is the geometric sensitivity  $G_1$ . As shown in [1] and detailed in section (3.5) it can be demonstrated that in some cases of interest this term equates to 0, simplifying considerably sensitivity evaluation.

# Chapter 3

## Force optimization

### 3.1 Problem definition

The reference problem layout is depicted in fig. 3.1. A 2D case is shown for better readability, yet the results presented in this study applies to both 2D and 3D flows. The set of BCs for the outer boundary is chosen among possible ones in order to include in the formulation a broader variety of BCs, instead of for instance a single far-field condition. A symmetry (or slip) boundary is introduced, it behaves like an impenetrable wall exerting no shear on the flow.

A multiple body configuration can be handled as well without any change in the adjoint setup, except if  $\Gamma_J$  and  $\Gamma_\beta$  are not coincident. This possibility can however be handled without much effort.

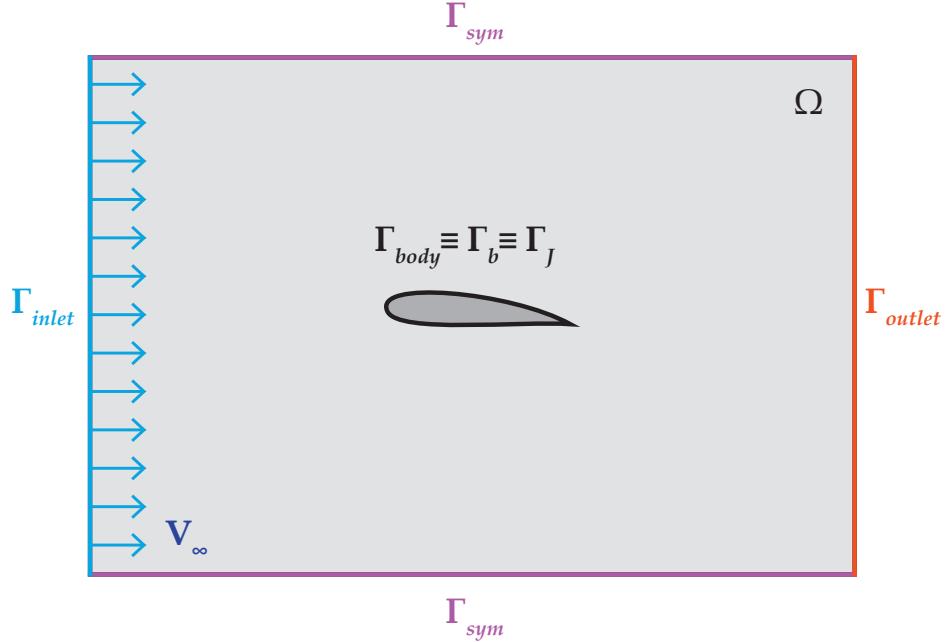
The approaches described in the previous section are now specialized starting from primal and adjoint fields

$$\mathbf{U} = [p, \mathbf{v}]^T, \quad \boldsymbol{\Psi} = [q, \mathbf{u}]^T \quad (3.1)$$

with  $q$  and  $\mathbf{u}$  named adjoint pressure and adjoint velocity. The primal problem is represented by steady incompressible Navier-Stokes equations

$$\nabla \cdot \mathbf{v} = 0 \quad \text{on } \Omega \quad (3.2)$$

$$\nabla \cdot (\mathbf{v}\mathbf{v}) + \nabla p - \nabla \cdot (2\nu\mathbb{D}(\mathbf{v})) = \mathbf{0} \quad \text{on } \Omega \quad (3.3)$$



**Figure 3.1:** Problem layout for external aerodynamics

wherein  $\mathbb{D}(\mathbf{v})$  represents the strain rate tensor, and from the following set of boundary conditions

$$\frac{\partial p}{\partial n} = 0, \quad \mathbf{v} = \mathbf{0} \quad \text{on } \Gamma_{body} \quad (3.4)$$

$$\frac{\partial p}{\partial n} = 0, \quad \mathbf{v} = V_\infty \mathbf{d}_\infty \quad \text{on } \Gamma_{inlet} \quad (3.5)$$

$$p = 0, \quad \frac{\partial \mathbf{v}}{\partial n} = \mathbf{0} \quad \text{on } \Gamma_{outlet} \quad (3.6)$$

$$\frac{\partial p}{\partial n} = 0, \quad v_n = 0, \quad \frac{\partial \mathbf{v}_t}{\partial n} = \mathbf{0} \quad \text{on } \Gamma_{sym} \quad (3.7)$$

wherein outlet and symmetry BCs are the approximated version of

$$p = 0, \quad \mathbf{n} \cdot \sigma = \mathbf{0} \quad \text{on } \Gamma_{outlet} \quad (3.8)$$

$$\frac{\partial p}{\partial n} = 0, \quad v_n = 0, \quad \mathbf{n} \cdot \sigma \cdot \mathbf{t}_1 = \mathbf{n} \cdot \sigma \cdot \mathbf{t}_2 = 0 \quad \text{on } \Gamma_{sym} \quad (3.9)$$

with  $\sigma$  representing the viscous stress tensor. Recall that the following property applies due to continuity equation

$$\frac{\partial v_n}{\partial n} = 0 \quad \text{on } \Gamma_{body} \cup \Gamma_{inlet} \quad (3.10)$$

Objective functions consist in the projection of the resultant force acting on the

body on a specified direction, given by the unit vector  $\mathbf{d}$

$$J = \int_{\Gamma_{body}} g(\mathbf{U}, \mathbf{n}, \mathbf{d}) \quad (3.11)$$

## 3.2 Pressure Force Formulation (PFF)

In this formulation, based on LBA,  $J$  includes only the resultant of pressure stresses

$$J = \frac{1}{C_\infty} \int_{\Gamma_{body}} -\mathbf{n} \cdot \sigma_p \cdot \mathbf{d} = \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot (p\mathbb{I}) \cdot \mathbf{d} \quad (3.12)$$

The minus sign is due to the normals' convention (exiting from  $\Omega$ ) and  $C_\infty$  is the reference used to non-dimensionalize force.

$$C_\infty = 1/2V_\infty^2 A \quad (3.13)$$

The template of sec. 2.1 is followed. with reference to (2.7), Lagrange function is defined

$$L(p, \mathbf{v}, q, \mathbf{u}) = J + \int_{\Omega} (qR_C(\mathbf{v}) + \mathbf{u} \cdot \mathbf{R}_M(p, \mathbf{v})) \quad (3.14)$$

with  $R_C$  and  $\mathbf{R}_M$  representing the differential operators of continuity and momentum equations, the first of the two is taken with minus sign. This is the convention followed by Othmer, the opposite choice will only affect the sign of adjoint pressure  $q$ . Considering from now on  $(p, \mathbf{v})$  solution of the primal equation on the undeformed boundary, the total variation of  $L$  reads

$$\delta L = \delta_\beta L + \delta_p L + \delta_{\mathbf{v}} L \quad (3.15)$$

With  $\delta\beta$  small. The variation w.r.t. flow variables is made to vanish originating the adjoint problem as in eq. (2.11)

$$\begin{aligned} & \text{find } (q, \mathbf{u}) : \\ & \delta_p J + \delta_{\mathbf{v}} J - \int_{\Omega} q \nabla \cdot \delta \mathbf{v} + \int_{\Omega} \mathbf{u} \cdot \{ \nabla \cdot (\mathbf{v} \delta \mathbf{v} + \delta \mathbf{v} \mathbf{v}) + \nabla \cdot (\delta p \mathbb{I} - 2\nu \mathbb{D}(\delta \mathbf{v})) \} = 0 \\ & \quad \forall \delta \mathbf{v}, \delta p \end{aligned} \quad (3.16)$$

Performing integration by parts and other manipulation

$$\int_{\Gamma_{body}} \delta_p g - \int_{\Omega} \nabla \cdot \mathbf{u} \delta p + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{u} \delta p = 0, \quad \forall \delta p \quad (3.17)$$

$$\begin{aligned} & \int_{\Gamma_{body}} \delta_v g - \int_{\Omega} \{\mathbf{v} \cdot \mathbb{D}(\mathbf{u}) - \nabla q + \nabla \cdot (2\nu \mathbb{D}(\mathbf{u}))\} \cdot \delta \mathbf{v} - \int_{\partial\Omega} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{u} \\ & + \int_{\partial\Omega} \{(\mathbf{v} \cdot \mathbf{n})\mathbf{u} + (\mathbf{v} \cdot \mathbf{u})\mathbf{n} - q\mathbf{n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u})\} \cdot \delta \mathbf{v} = 0 \quad \forall \delta \mathbf{v} \end{aligned} \quad (3.18)$$

Remark: if the convection term is kept in conservation form the assumption  $\nabla \cdot \delta \mathbf{v} = 0$  can be avoided.

Adjoint equation are extracted

$$\nabla \cdot \mathbf{u} = \mathbf{0} \quad \text{on } \Omega \quad (3.19)$$

$$-\mathbf{v} \cdot 2\mathbb{D}(\mathbf{u}) + \nabla q - \nabla \cdot (2\nu \mathbb{D}(\mathbf{u})) = \mathbf{0} \quad \text{on } \Omega \quad (3.20)$$

And BCs are obtained in weak form as

$$\int_{\Gamma_{body}} \delta_p g + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{u} \delta p = 0 \quad \forall \delta p \quad (3.21)$$

$$\begin{aligned} & \int_{\Gamma_{body}} \delta_v g - \int_{\partial\Omega} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{u} \\ & + \int_{\partial\Omega} \{(\mathbf{v} \cdot \mathbf{n})\mathbf{u} + (\mathbf{v} \cdot \mathbf{u})\mathbf{n} - q\mathbf{n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u})\} \cdot \delta \mathbf{v} = 0 \quad \forall \delta \mathbf{v} \end{aligned} \quad (3.22)$$

Resorting to Equation 26 of [5], reported here:

$$\begin{aligned} & \int_{\partial\Omega} 2\nu \mathbf{n} \cdot \{\mathbb{D}(\mathbf{u}) \cdot \delta \mathbf{v} - \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{u}\} \\ & = \int_{\partial\Omega} \nu \left\{ \frac{\partial \mathbf{u}}{\partial n} \cdot \delta \mathbf{v} - \frac{\partial \delta \mathbf{v}}{\partial n} \cdot \mathbf{u} \right\} - \int_{\partial\Omega} \nabla \nu \cdot \{u_n \delta \mathbf{v} - \delta v_n \mathbf{u}\} \end{aligned} \quad (3.23)$$

if  $\nabla \nu = 0$  the terms containing operator  $\mathbb{D}(.)$  can be simplified, leading to

$$\int_{\Gamma_{body}} \delta_v g - \int_{\partial\Omega} \nu \frac{\partial \delta \mathbf{v}}{\partial n} \cdot \mathbf{u} + \int_{\partial\Omega} \left\{ (\mathbf{v} \cdot \mathbf{n})\mathbf{u} + (\mathbf{v} \cdot \mathbf{u})\mathbf{n} - q\mathbf{n} + \nu \frac{\partial \mathbf{u}}{\partial n} \right\} \cdot \delta \mathbf{v} = 0 \quad (3.24)$$

This is always the case for a laminar flow, while an error is introduced on the outlet patch in the case of turbulent flow.

Except for the symmetry patch, the other BCs are derived exactly as in [5] and will be only listed. Worth of a mention is the handling of adjoint pressure, whose BCs are not completely defined by eqs. (3.21) and (3.24). Given the analogy

between primal and adjoint equation, if a predictor-corrector scheme is used on the adjoint problem as well,  $q$  can be treated the same way  $p$  is. This translates into a zero normal gradient condition for  $q$  wherever the correction to boundary faces' adjoint fluxes is forced to be zero.

### body

$$\frac{\partial q}{\partial n} = 0, \quad u_n = -\frac{\partial g}{\partial p} = -\frac{\mathbf{n} \cdot \mathbf{d}}{C_\infty}, \quad \mathbf{u}_t = \mathbf{0} \quad (3.25)$$

### inlet

$$\frac{\partial q}{\partial n} = 0, \quad \mathbf{u} = \mathbf{0} \quad (3.26)$$

### outlet

$$q = \mathbf{u} \cdot \mathbf{v} + u_n v_n + \nu \frac{\partial u_n}{\partial n}, \quad v_n \mathbf{u}_t + \nu \frac{\partial \mathbf{u}_t}{\partial n} = \mathbf{0} \quad (3.27)$$

### symmetry (new)

From eq.(3.21) follows

$$u_n = 0$$

This means for consistency with the pressure corrector equation

$$\frac{\partial q}{\partial n} = 0$$

Considering eq.(3.24): the first integral is zero by definition, the integrand of the second one vanishes being the inner product of orthogonal vectors, since  $\partial_n \mathbf{v}_t = \mathbf{0}$  and  $u_n = 0$ , finally further substituting  $v_n = \delta v_n = 0$  the expression is reduced to

$$\int_{\Gamma_{slip}} \nu \frac{\partial \mathbf{u}_t}{\partial n} \cdot \mathbf{v}_t = 0$$

leading to

$$\frac{\partial \mathbf{u}_t}{\partial n} = \mathbf{0}$$

One can notice the dual BC collected in eq. (3.28) is again a symmetry condition

$$\frac{\partial q}{\partial n} = 0, \quad u_n = 0, \quad \frac{\partial \mathbf{u}_t}{\partial n} = \mathbf{0} \quad (3.28)$$

The expression of sensitivity is now derived. Manipulating  $\delta_\beta L$  the metrics-free variant (2.27) is obtained

$$\begin{aligned}\delta_\beta L &= \delta_\beta J - \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{u} \delta p + \int_{\partial\Omega} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{u} \\ &\quad - \int_{\partial\Omega} \{(\mathbf{v} \cdot \mathbf{n})\mathbf{u} + (\mathbf{v} \cdot \mathbf{u})\mathbf{n} - q\mathbf{n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u})\} \cdot \delta \mathbf{v}\end{aligned}\tag{3.29}$$

The geometric term will be addressed first: applying the divergence theorem to recast eq. (2.22) into a volume integral and handling its variation w.r.t. (small) boundary perturbation like in Reynolds transport theorem, the following is obtained

$$\delta_\beta J = \frac{1}{C_\infty} \int_{\Gamma_{body}} \delta \beta (\nabla p \cdot \mathbf{d})\tag{3.30}$$

One can notice  $p$  appears in this term with an higher order of differentiation than in the state vector, lower accuracy is to be expected. Considering now the remaining part of (3.29), and applying primal and adjoint BCs

$$\frac{1}{C_\infty} \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{d} \delta p - \frac{1}{C_\infty} \int_{\partial\Omega} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{n}(\mathbf{n} \cdot \mathbf{d}) - \int_{\partial\Omega} \{-q\mathbf{n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u})\} \cdot \delta \mathbf{v}\tag{3.31}$$

Differently from the case treated by Othmer, in which  $\mathbf{u}$  vanishes on the wall, the present doesn't allow the simplification introduced in eqs. 69 and 72 of [5], hence

$$2\mathbf{n} \cdot \mathbb{D}(\mathbf{u}) \neq \frac{\partial \mathbf{u}}{\partial n}\tag{3.32}$$

Next, eq. (2.26) specializes to

$$\delta p \simeq \frac{\partial p}{\partial n} \delta \beta, \quad \delta \mathbf{v} \simeq \frac{\partial \delta \mathbf{v}}{\partial n} \delta \beta \quad \text{on } \Gamma_{body}\tag{3.33}$$

Substituting the approximated  $\delta \mathbf{v}$  and  $\delta p$  into (3.31)

$$\begin{aligned}&\frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot \mathbf{d} \frac{\partial p}{\partial n} \delta \beta - \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot 2\nu \mathbb{D} \left( \frac{\partial \mathbf{v}_t}{\partial n} \delta \beta \right) \cdot \mathbf{n}(\mathbf{n} \cdot \mathbf{d}) \\ &- \int_{\Gamma_{body}} \left\{ -q \frac{\partial v_n}{\partial n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u}) \cdot \frac{\partial \mathbf{v}_t}{\partial n} \right\} \delta \beta\end{aligned}\tag{3.34}$$

By virtue of  $\partial_n v_n$  and  $\partial_n p$  being zero and performing additional manipulation to

(3.34) the final expression of sensitivity is obtained

$$\begin{aligned} \delta_\beta L = & \frac{1}{C_\infty} \int_{\Gamma_{body}} (\nabla p \cdot \mathbf{d}) \delta\beta - \frac{1}{C_\infty} \int_{\Gamma_{body}} 2\nu \frac{\partial^2 v_n}{\partial n^2} (\mathbf{n} \cdot \mathbf{d}) \delta\beta \\ & - \int_{\Gamma_{body}} \left\{ \nu \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} + \nu \mathbf{n} \cdot \nabla^T \mathbf{u} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \right\} \delta\beta \end{aligned} \quad (3.35)$$

Finally collecting surface sensitivity

$$G = \frac{1}{C_\infty} \left( \nabla p \cdot \mathbf{d} - 2\nu \frac{\partial^2 v_n}{\partial n^2} (\mathbf{n} \cdot \mathbf{d}) \right) - \nu \left( \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} + \mathbf{n} \cdot \nabla^T \mathbf{u} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \right) \quad (3.36)$$

The resulting expression is pretty articulated and involves two higher order derivative terms.

### 3.3 Lagrangian-based Total Force Formulation (TFF<sub>LBA</sub>)

In this formulation, based on LBA,  $J$  includes both pressure and shear contributions

$$J = \frac{1}{C_\infty} \int_{\Gamma_{body}} -\mathbf{n} \cdot \sigma_{tot} \cdot \mathbf{d} = \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot (p\mathbb{I} - \sigma) \cdot \mathbf{d} \quad (3.37)$$

The addition of the viscous term in the objective function will affect only the adjoint boundary condition on the wall and sensitivity, leaving the rest as in sec. 3.2. Let's deduce the new BCs evaluating eq. (3.22) on the body surface. The third integral cancels by virtue of  $\delta\mathbf{v}$  being zero, then expanding the first

$$\frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta\mathbf{v}) \cdot \mathbf{d} + \int_{\Gamma_{body}} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta\mathbf{v}) \cdot \mathbf{u} = 0 \quad (3.38)$$

Integrals can be made to cancel each other if the following BC is applied

$$\mathbf{u} = -\frac{1}{C_\infty} \mathbf{d} \quad \text{on } \Gamma_{body} \quad (3.39)$$

This is finally completed with

$$\frac{\partial q}{\partial n} = 0 \quad \text{on } \Gamma_{body} \quad (3.40)$$

Let's consider now sensitivity: substituting (3.33) into (3.29) and applying primal and adjoint BCs, we get

$$\delta_\beta L = \delta_\beta J - \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot 2\nu \mathbb{D} \left( \frac{\partial \mathbf{v}_t}{\partial n} \delta \beta \right) \cdot \mathbf{d} - \int_{\Gamma_{body}} \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u}) \cdot \frac{\partial \mathbf{v}_t}{\partial n} \delta \beta \quad (3.41)$$

Let's take care first of geometric term: using the same procedure followed in the derivation of eq. (3.30) the integral can be conveniently reformulated as

$$\int_{\delta \Gamma_{body}} \mathbf{n} \cdot [p\mathbb{I} - \sigma] \cdot \mathbf{d} = \int_{\Gamma_{body}} \delta \beta (\nabla \cdot [p\mathbb{I} - \sigma]) \cdot \mathbf{d} \quad (3.42)$$

Proceeding as suggested by Castro et al. one can show the term in round brackets equates to momentum equation evaluated at the wall and hence

$$\nabla \cdot [p\mathbb{I} - \sigma] = \mathbf{0} \quad \text{on } \Gamma_{body} \quad (3.43)$$

We can conclude geometric term vanishes, gaining us the elision of some higher derivative terms, otherwise present. Finally performing some algebra on the remaining part of (3.41) the final expression is obtained

$$\begin{aligned} \delta_\beta L &= -\frac{1}{C_\infty} \int_{\Gamma_{body}} \nu \left\{ \frac{\partial^2 \mathbf{v}}{\partial n^2} \cdot \mathbf{d} + \mathbf{n} \cdot \nabla^T \mathbf{v} \cdot \frac{\partial \mathbf{n}}{\partial d} + \frac{\partial^2 \mathbf{v}}{\partial n \partial d} \cdot \mathbf{n} \right\} \delta \beta \\ &\quad - \int_{\Gamma_{body}} \nu \left\{ \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} + \mathbf{n} \cdot \nabla^T \mathbf{u} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \right\} \delta \beta \end{aligned} \quad (3.44)$$

And in terms of local sensitivity

$$\begin{aligned} G &= -\frac{\nu}{C_\infty} \left( \frac{\partial^2 \mathbf{v}}{\partial n^2} \cdot \mathbf{d} + \mathbf{n} \cdot \nabla^T \mathbf{v} \cdot \frac{\partial \mathbf{n}}{\partial d} + \frac{\partial^2 \mathbf{v}}{\partial n \partial d} \cdot \mathbf{n} \right) \\ &\quad - \nu \left( \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} + \mathbf{n} \cdot \nabla^T \mathbf{u} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \right) \end{aligned} \quad (3.45)$$

As in the previous section sensitivity is not higher order derivatives free.

## 3.4 Momentum Balance Formulation (MBF)

This formulation, based on LBA, was born as a different point of view on cost function definition and with the purpose of simplifying the evaluation of sensitivity. The point is computing the force exerted on the body resorting to the momentum balance on the computational domain, instead of integrating wall stresses. In a

steady flow the forces exerted on the fluid throughout the boundary of a control volume must equate the net outflow of momentum through the same surfaces. Because of the non-slip condition, only forces are exerted on the wall boundary, so they can be computed indirectly by evaluating the balance integrals on the outer boundary. Let's right multiply the balance relative to eq. (3.3) by the projection direction  $\mathbf{d}$ :

$$\int_{\partial\Omega} \mathbf{v}(\mathbf{v} \cdot \mathbf{n}) \cdot \mathbf{d} = \int_{\partial\Omega} \mathbf{n} \cdot [\sigma - p\mathbb{I}] \cdot \mathbf{d} \quad (3.46)$$

With reference to our problem layout the balance reduces to

$$\begin{aligned} & \int_{\Gamma_{inlet}} V_\infty^2 (\mathbf{d}_\infty \cdot \mathbf{n})(\mathbf{d}_\infty \cdot \mathbf{d}) + \int_{\Gamma_{outlet}} (\mathbf{v} \cdot \mathbf{n})(\mathbf{v} \cdot \mathbf{d}) \\ &= -\mathbf{R}_b \cdot \mathbf{d} - \int_{\Gamma_{inlet}} p(\mathbf{n} \cdot \mathbf{d}) - \int_{\Gamma_{sym}} p(\mathbf{n} \cdot \mathbf{d}) \end{aligned} \quad (3.47)$$

wherein the resultant force exerted on the body has been collected into  $\mathbf{R}_b$ , normal viscous stresses neglected on inlet and symmetry boundaries and other terms canceled due to BCs. Considering now  $J$  defined as in eq. (3.37) and obtaining the term  $\mathbf{R}_b \cdot \mathbf{d}$  from eq. (3.47) we can express the objective function as an integral on the outer boundary

$$\begin{aligned} C_\infty J = & - \int_{\Gamma_{inlet}} \{V_\infty^2 (\mathbf{d}_\infty \cdot \mathbf{n})(\mathbf{d}_\infty \cdot \mathbf{d}) + p(\mathbf{n} \cdot \mathbf{d})\} \\ & - \int_{\Gamma_{outlet}} (\mathbf{v} \cdot \mathbf{n})(\mathbf{v} \cdot \mathbf{d}) - \int_{\Gamma_{sym}} p(\mathbf{n} \cdot \mathbf{d}) \end{aligned} \quad (3.48)$$

The expression above can be further simplified in the case of box shaped outer boundary and parallel wind.

The main advantage of this definition is having complementary  $\Gamma_J$  and  $\Gamma_b$ , in other words control and cost function are defined on non overlapping portion of the boundary. Remark: this formulation doesn't require force coefficients to be evaluated using momentum balance, yet care must be taken in solving primal equation with sufficient accuracy. The validity of eq. (3.47) relies in fact on momentum balance being satisfied on the computational domain, this translates at discrete level in residuals on momentum equations being small enough. For this reason a finer degree of convergence should be achieved in solving primal equations.

Let's now focus on the changes introduced by this alternative definition of  $J$ . Boundary conditions are deduced as usual from eqs. (3.21) and (3.24):

**body** On this boundary the dependence on the cost function is lost, simplifying BCs to

$$\frac{\partial q}{\partial n} = 0, \quad \mathbf{u} = \mathbf{0} \quad (3.49)$$

**inlet** The term  $\delta_p g$  differs from zero in eq. (3.21), yielding

$$\frac{\partial q}{\partial n} = 0, \quad u_n = \frac{1}{C_\infty} d_n, \quad \mathbf{u}_t = \mathbf{0} \quad (3.50)$$

**outlet** On the outlet, BCs differs from (3.27) only for the addition of the term  $\delta_v g$ , which is found decomposed into normal and tangent directions

$$\begin{aligned} q &= \mathbf{u} \cdot \mathbf{v} + u_n v_n + \nu \frac{\partial u_n}{\partial n} - \frac{1}{C_\infty} (\mathbf{v} \cdot \mathbf{d} + v_n d_n) \\ v_n \mathbf{u}_t + \nu \frac{\partial \mathbf{u}_t}{\partial n} &= v_n \mathbf{d}_t \end{aligned} \quad (3.51)$$

**symmetry** The term  $\delta_p g$  differs from zero in eq. (3.21), yielding

$$\frac{\partial q}{\partial n} = 0, \quad u_n = \frac{1}{C_\infty} d_n, \quad \frac{\partial \mathbf{u}_t}{\partial n} = \mathbf{0} \quad (3.52)$$

wherein  $d_n$  and  $\mathbf{d}_t$  represent respectively surface normal and surface tangent decomposition of  $\mathbf{d}$ .

Next, sensitivity is derived by evaluating eq. (3.29) on the control boundary  $\Gamma_b$ . Several simplifications are possible: since the objective functional is not affected by control, geometric term vanishes, then second and third integrals equate to zero because of the “no-slip” condition on  $\mathbf{u}$ , finally the last integral is simplified in a similar manner and the term in  $q$  cancels as soon as the variation  $\delta \mathbf{v}$  from (3.33) is substituted. The final expression is simply

$$\delta_\beta L = - \int_{\Gamma_{body}} \nu \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \quad (3.53)$$

It can be noted the expression is the same as the one derived by Othmer for ducted flows. This alternative formulation is indeed not dissimilar from the one previously mentioned, exception be made for the duct being the “numerical wind tunnel” and shape optimization performed on an immersed obstacle.

### 3.5 Lagrangian-free Total Force Formulation (TFF<sub>LFA</sub>)

This formulation is the LFA variant of sec. 3.3, hence

$$J = \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot (p\mathbb{I} - \sigma) \cdot \mathbf{d} \quad (3.37)$$

Its total variation reads

$$\delta J = \frac{1}{C_\infty} \int_{\delta\Gamma_{body}} \mathbf{n} \cdot (p\mathbb{I} - \sigma) \cdot \mathbf{d} + \frac{1}{C_\infty} \int_{\Gamma_{body}} \mathbf{n} \cdot (\delta p\mathbb{I} - \delta\sigma) \cdot \mathbf{d} \quad (3.54)$$

From this point forward the procedure in sec. 2.2 is followed. Primal equations are linearized:

$$\nabla \cdot \delta\mathbf{v} = \mathbf{0} \quad \text{on } \Omega \quad (3.55)$$

$$\nabla \cdot (\mathbf{v}\delta\mathbf{v} + \delta\mathbf{v}\mathbf{v} + \delta\mathbf{p}\mathbb{I} - 2\nu\mathbb{D}(\delta\mathbf{v})) = \mathbf{0} \quad \text{on } \Omega \quad (3.56)$$

As in the reference case typically primal BCs are linear, this allows eq. (2.31) to be simplified, yielding

$$\delta\mathbf{B}(p, \mathbf{v}) \stackrel{lin}{=} \mathbf{B}(\delta p, \delta\mathbf{v}) = \mathbf{0} \quad \text{on } \partial\Omega \quad (3.57)$$

This means that variation on the outer boundary will satisfy the BCs of the primal problem, the only exception being, strictly speaking, the inhomogeneous Dirichlet condition on inlet, that will be made homogeneous by linearization; on body surface instead eq. (2.38) applies

$$\mathbf{B}(\delta p, \delta\mathbf{v}) = \mathbf{0} \quad \text{on } \partial\Omega \setminus \Gamma_{body} \quad (3.58)$$

$$\frac{\partial \delta p}{\partial n} = -\delta\beta \frac{\partial^2 p}{\partial n^2}, \quad \delta\mathbf{v} = -\delta\beta \frac{\partial \mathbf{v}}{\partial n} \quad \text{on } \Gamma_{body} \quad (3.59)$$

Notice that the latter term of eq. (3.59) is identical except in sign to the respective of eq. (3.33). The weak form of linearized flow equations is obtained

$$\int_{\Omega} -q\nabla \cdot \delta\mathbf{v} + \int_{\Omega} \mathbf{u} \cdot \{\nabla \cdot (\mathbf{v}\delta\mathbf{v} + \delta\mathbf{v}\mathbf{v}) + \nabla \cdot (\delta p\mathbb{I} - 2\nu\mathbb{D}(\delta\mathbf{v}))\} = 0 \\ \forall \text{ admissible } \delta\mathbf{v}, \delta p \quad (3.60)$$

The minus sign in the first integral is introduced, following the convention of sec. 2.1. The weak form of the adjoint problem is obtained in the same form of eq.

(2.16).

$$\int_{\Omega} \nabla \cdot \mathbf{u} \delta p - \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{u} \delta p = 0, \quad \forall \text{ admissible } \delta p \quad (3.61)$$

$$\begin{aligned} & - \int_{\Omega} \{ \mathbf{v} \cdot \mathbb{D}(\mathbf{u}) - \nabla q + \nabla \cdot (2\nu \mathbb{D}(\mathbf{u})) \} \cdot \delta \mathbf{v} - \int_{\partial\Omega} \mathbf{n} \cdot 2\nu \mathbb{D}(\delta \mathbf{v}) \cdot \mathbf{u} \\ & + \int_{\partial\Omega} \{ (\mathbf{v} \cdot \mathbf{n}) \mathbf{u} + (\mathbf{v} \cdot \mathbf{u}) \mathbf{n} - q \mathbf{n} + \mathbf{n} \cdot 2\nu \mathbb{D}(\mathbf{u}) \} \cdot \delta \mathbf{v} = 0 \\ & \quad \forall \text{ admissible } \delta \mathbf{v} \end{aligned} \quad (3.62)$$

Adjoint equation are collected:

$$\nabla \cdot \mathbf{u} = \mathbf{0} \quad \text{on } \Omega \quad (3.19)$$

$$-\mathbf{v} \cdot 2\mathbb{D}(\mathbf{u}) + \nabla q - \nabla \cdot (2\nu \mathbb{D}(\mathbf{u})) = \mathbf{0} \quad \text{on } \Omega \quad (3.20)$$

The remaining boundary terms account to

$$\begin{aligned} & \int_{\partial\Omega} \{ (\mathbf{v} \cdot \mathbf{n}) \mathbf{u} + (\mathbf{v} \cdot \mathbf{u}) \mathbf{n} + \mathbf{n} \cdot [-q \mathbb{I} + 2\nu \mathbb{D}(\mathbf{u})] \} \cdot \delta \mathbf{v} \\ & - \int_{\partial\Omega} \mathbf{n} \cdot [-\delta p \mathbb{I} + 2\nu \mathbb{D}(\delta \mathbf{v})] \cdot \mathbf{u} = 0, \quad \forall \text{ admissible } \delta p, \delta \mathbf{v} \end{aligned} \quad (3.63)$$

wherein the terms in square brackets can be recognized respectively as the adjoint (or dual) total stress tensor and the variation of the total stress tensor. It must be stressed that the second statement is true if  $\delta\nu = 0$ , that's the case of laminar flows (or of frozen turbulence approximation). For symmetry of notation we define  $\Sigma := 2\nu \mathbb{D}(\mathbf{u})$ .

In the following, boundary integrals are kept in the form of (3.63) and the non-approximated version of outflow and symmetry BCs will be used, i.e. eqs. (3.8), (3.9). Integrals will be evaluated on each patch separately, starting from the outer boundary:

**inlet** upon substitution of  $\delta \mathbf{v} = \mathbf{0}$  the integrals on the inlet patch reduce to

$$\int_{\Gamma_{inlet}} \mathbf{n} \cdot [\delta p \mathbb{I} - \delta \sigma] \cdot \mathbf{u} = 0, \quad \forall \text{ admissible } \delta p, \delta \sigma \quad (3.64)$$

The expression vanishes for

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{inlet} \quad (3.65)$$

Handling adjoint pressure in the same way of previous sections, this condition

will be complemented with

$$\frac{\partial q}{\partial n} = 0 \quad \text{on } \Gamma_{inlet} \quad (3.66)$$

**outlet** On the outlet patch  $\delta p = 0$  and  $\mathbf{n} \cdot \delta \sigma = \mathbf{0}$ , hence the right integral of (3.63) cancels

$$\int_{\Gamma_{outlet}} \{(\mathbf{v} \cdot \mathbf{n})\mathbf{u} + (\mathbf{v} \cdot \mathbf{u})\mathbf{n} + \mathbf{n} \cdot [-q\mathbb{I} + 2\nu\mathbb{D}(\mathbf{u})]\} \cdot \delta \mathbf{v} = 0 \quad (3.67)$$

By equating to zero the integrand and decomposing it in normal and tangent directions the following BCs are derived respectively

$$\begin{aligned} q &= \mathbf{u} \cdot \mathbf{v} + u_n v_n + 2\nu \frac{\partial u_n}{\partial n} \\ v_n \mathbf{u}_t + \nu \left( \frac{\partial u_n}{\partial t_1} + \frac{\partial u_n}{\partial t_2} + \frac{\partial \mathbf{u}_t}{\partial n} \right) &= \mathbf{0} \quad \text{on } \Gamma_{outlet} \end{aligned} \quad (3.68)$$

**symmetry** On this boundary  $\delta p \neq 0$ , hence equating to zero the respective term the following is deduced

$$u_n = 0, \quad \text{on } \Gamma_{sym} \quad (3.69)$$

Applying the usual treatment of adjoint pressure it follows

$$\frac{\partial q}{\partial n} = 0, \quad \text{on } \Gamma_{sym} \quad (3.70)$$

By vanishing  $u_n$ ,  $\delta v_n$  and  $v_n$  the second integral is canceled and the first one simplified to

$$\int_{\Gamma_{sym}} \mathbf{n} \cdot \boldsymbol{\Sigma} \cdot \delta \mathbf{v}_t = 0 \quad (3.71)$$

This condition translates into

$$\mathbf{n} \cdot \boldsymbol{\Sigma} \cdot \mathbf{t}_1 = \mathbf{n} \cdot \boldsymbol{\Sigma} \cdot \mathbf{t}_2 = 0 \quad \text{on } \Gamma_{sym} \quad (3.72)$$

As found for its approximated version, the primal symmetry condition has a specular counterpart in the adjoint problem.

**body** Finally on the body patch, substituting  $\mathbf{v} = \mathbf{0}$

$$\int_{\Gamma_{body}} \mathbf{n} \cdot [-q\mathbb{I} + \Sigma] \cdot \delta\mathbf{v} + \int_{\Gamma_{body}} \mathbf{n} \cdot [\delta p\mathbb{I} - \delta\sigma] \cdot \mathbf{u} = 0$$

$$\forall \text{ admissible } \delta p, \delta\sigma \quad (3.73)$$

It can be noticed that the right integral is very similar to the field term of sensitivity (3.54) and can be made equal if the following BC is assigned to  $\mathbf{u}$

$$\mathbf{u} = \frac{1}{C_\infty} \mathbf{d}, \quad \text{on } \Gamma_{body} \quad (3.74)$$

The expression of this component of sensitivity is then available

$$\int_{\Gamma_{body}} \frac{\partial g}{\partial U} \delta\mathbf{U} = \int_{\Gamma_{body}} \mathbf{n} \cdot [q\mathbb{I} - \Sigma] \cdot \delta\mathbf{v} \quad (3.75)$$

Finally being  $\mathbf{u}$  prescribed

$$\frac{\partial q}{\partial n} = 0 \quad \text{on } \Gamma_{body} \quad (3.76)$$

Remark: in the case of laminar flow eqs. (3.27) and (3.28) can be used instead of the present outflow and symmetry BCs without committing error.

The missing component of sensitivity, i.e. geometric term, equates to zero for the present objective function as shown in (3.42-3.43). This means the full expression of sensitivity is given by eq. (3.75). Finally substituting  $\delta\mathbf{v}$  from eq. (3.59) into the latter, the dependence on control is made explicit

$$\delta J = \int_{\Gamma_{body}} \left( \mathbf{n} \cdot [-q\mathbb{I} + \Sigma] \cdot \frac{\partial \mathbf{v}}{\partial n} \right) \delta\beta \quad (3.77)$$

Using (3.10) and (3.74) surface sensitivity, i.e. the expression in round brackets, can be simplified yielding

$$G = \nu \frac{\partial \mathbf{u}_t}{\partial n} \cdot \frac{\partial \mathbf{v}_t}{\partial n} \quad (3.78)$$

The reduction of higher derivative terms dealt in sec. III of [1] has been successfully reproduced. This result enhances greatly the accuracy of the sensitivity for cases solved with conventional FV schemes.

## 3.6 Implementation in OpenFOAM

OpenFOAM is an Open Source C++ toolbox suitable for the development of customized numerical solvers. It features professional level CFD solvers and a number pre-/post-processing utilities. The formulations presented in this study have been packaged into solvers, each with its specific set of BCs and post-processing of sensitivity. Another possibility, not pursued at present, was to compile a single adjoint solver and complement it with custom BCs, each for a given cost function type. The same could be done for sensitivity, suitable to be implemented into a *object function*, to be either dynamically linked to the solver at run time or executed in post-processing.

### 3.6.1 Solvers

#### **adjointOptFoam v2**

The starting point was *adjointOptFoam* solver, developed in [3] and based on the native *adjointShapeOptimizationFoam*. The code, featuring a pressure force formulation, has been revised and improved updating BCs and sensitivity with the results of the present PFF.

#### **adjointDefectFoam**

This solver represents the intermediate step towards the next one. It implements a sub-case of Momentum Balance Formulation, the one obtained considering only velocity terms in the objective function. Specifically momentum defect on the outflow boundary is (ideally) used to compute  $C_D$  and so to define  $J$ . This variant has been discontinued, as it is based on the assumption momentum defect is evaluated on the far wake, hence producing large errors if the extension of the computational domain is not sufficient. A possible, yet unpractical alternative is to employ the Squire-Young formula for drag estimation of airfoils, which is of course of limited application and would require the definition of the cost function on a contour that is not part of any boundary.

#### **adjointBalanceFoam**

This solver overcomes the drawback of the previous implementing the full MBF. As in the previous case the required new BCs and sensitivity post-processing script have been obtained by modifying the original solver accordingly.

## adjointFullFoam

This was the last solver to be coded and features the Total Force Formulation-LFA. The Lagrangian-Free Approach, has been preferred over the Lagrangian-Based Approach (LBA) because it features more elegant and simple results.

### 3.6.2 Porting from v3.0 to v4.1

The code has been updated to the last OF version. The main changes are reported in the following.

#### Code

New management of reference types. Particularly for the *GeometricField* class:

- Reference to a *internalField*: *FieldName()* → *FieldName.ref()*
- Reference to a *boundaryField*: *FieldName.boundaryField()* → *FieldName.boundaryFieldRef()*

#### Case directory

- The *RASProperties* dictionary is now embedded in *turbulenceProperties* dictionary.
- *BlockMeshDict* has been moved from */constant/polyMesh* to */system*

### 3.6.3 Solver's features

In the following the salient features of the code are listed.

#### Standalone adjoint solution

As opposed to the native solver, featuring the solution of both primal and adjoint systems during each iteration, the present solves uniquely the adjoint problem using converged primal fields, which need to be provided as initial conditions.

#### SIMPLE-based solution algorithm

Considering the analogy between the two set of equations, it's common practice to employ for the adjoint problem the same method used to solve the primal. The choice falls in our case on the SIMPLE algorithm, which is the most popular

segregated predictor-corrector algorithm. The code is directly derived from the *simpleFoam* solver, which is modified taking advantage of the high level programming provided by OpenFOAM.

### Explicit Adjoint Transpose Convection

Being in the context of a fully segregated algorithm, solving for each velocity component at time, the ATC term is discretized explicitly. Specifically it's a source term built using the adjoint velocity from previous iteration and the fixed velocity field taken from initial conditions. The advantage of keeping the same algorithm is paid with a discretization that is more incline to instability. A range of stabilizing strategies are possible, for instance near wall damping of ATC or cell-wise damping for diagonal dominance preservation. See for more details section 2.3 of [12].

As consequence of the explicit updates present in equations and of usually stronger under-relaxation, adjoint solution may require a lot more iterations to converge if compared with the primal. Hence, despite being a linear PDE, the adjoint problem requires possibly more cpu time.

# Chapter 4

## Study on 2D flows

With the purpose of investigating the behaviour of the formulations herein described a study on 2D laminar flows has been carried. The restriction to laminar regime is thought to avoid the additional error source represented by turbulence modeling and the contamination of lesser than 2nd order accurate schemes. Among others, the subsonic test-cases found in [1] have been reproduced, being in fact consistent with our setting and providing a good reference for comparing results. The details of the latter test cases are summarized below:

- A)** Flow around a cylinder at Reynolds = 50
- B)** Flow past a NACA 0012 with an AoA of 2.5 deg and a Reynolds = 1000

### 4.1 Computational mesh

Since, as well known, the adjoint solution is very sensitive to the resolution and regularity of the computational grid, particular attention has been paid in mesh design. Both structured (*blockMesh*) and unstructured (*snappyHexMesh*) meshes have been tested, opting for the first, which are the best choice for simple 2D geometries. Subsequently, different layouts and settings were explored in setting up the input to the *blockMesh* mesh generator. Finally, with a view to obtaining better refinement and regularity control, a structured C mesh generator (*SCMG*) was implemented by using the language and libraries of *GNU Octave*. The software is based on the use of *cubic Hermite splines* for parameterizing the input geometry and *Bezier curves* to control the vertices layout. The mesh design, thought for airfoils, is freely inspired to the well-known set of grids developed by NASA for the NACA 0012 airfoil [6].

### 4.1.1 snappyHexMesh

*SnappyHexMesh* is the native unstructured hexaedral mesh generator in the current OF release. A mesh around a generic surface geometry can be generated following this workflow:

- The surface of the body to be meshed must be generated in a CAD format or imported.
- A structured mesh of hexaedra (usually cubes) is generated with *blockMesh*. This grid is known as “background mesh” and will define the overall dimensions of the final mesh. It must be set in such away that the body surface could fit into its bounds. Additionally the number of hexaedra will reflect on the number of cells of the final mesh.
- A “castellated” mesh is generated in the first step of *snappyHexMesh* execution. It consists in the original background mesh spoiled of any of the hexaedra that are either inside or outside the specified surface geometry. This requires the body surface and an point inside the flow domain to be input.

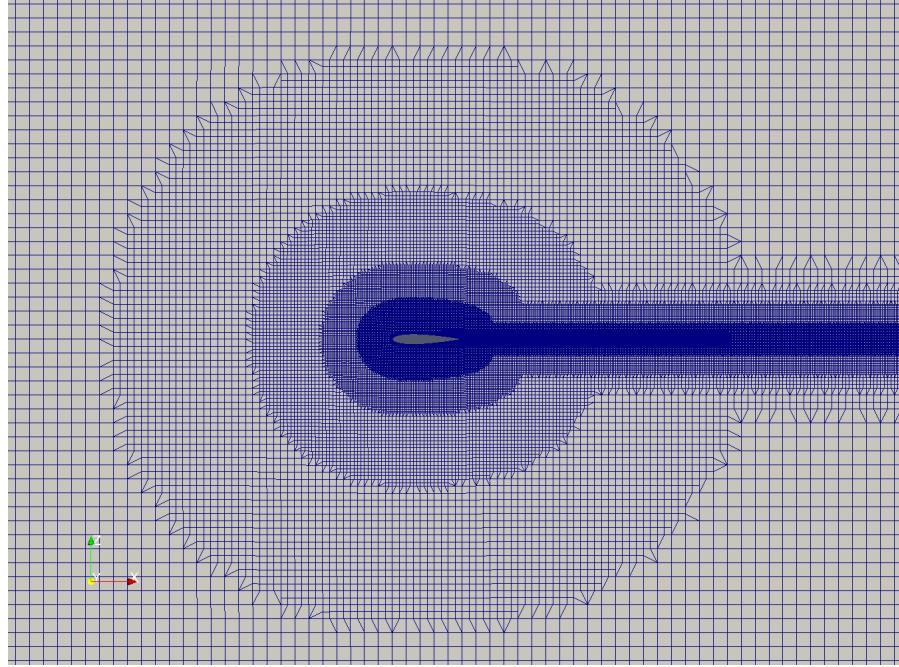
Before the elimination procedure a refinement phase occurs, which is based on a successive splitting of the hexaedra according to settings specified in *snappyHexMeshDict* dictionary. The splitting procedure can be applied to a subset of the cells and iterated multiple times according to the *refinementLevel* parameter.

- The next stage referred to as “snapping” involves moving cell vertex points onto surface geometry so to obtain a smooth boundary that conforms to the input geometry.
- The last (optional stage) is the additions of layers of cells to the body patch.

For the first step the free Matlab/Octave script *NACA2STL*, available on *GitHub* [18], has been used. The meshing application can handle only 3D meshes, for this reason an additional step must be taken in order to obtain a 2D one:

- Using *extrudeMesh* OF utility the “front” or “back” patches of the 3D mesh can be extruded to a specified length so to obtain a one cell thick unstructured mesh.

One of the unstructured meshes produced with this procedure is shown in fig. 4.1. The program was initially used for the execution of some tests but, due to the long processing time required, was abandoned right away.



**Figure 4.1:** NACA 0012 mesh generated with *snappyHexMesh* - far view

### 4.1.2 blockMesh

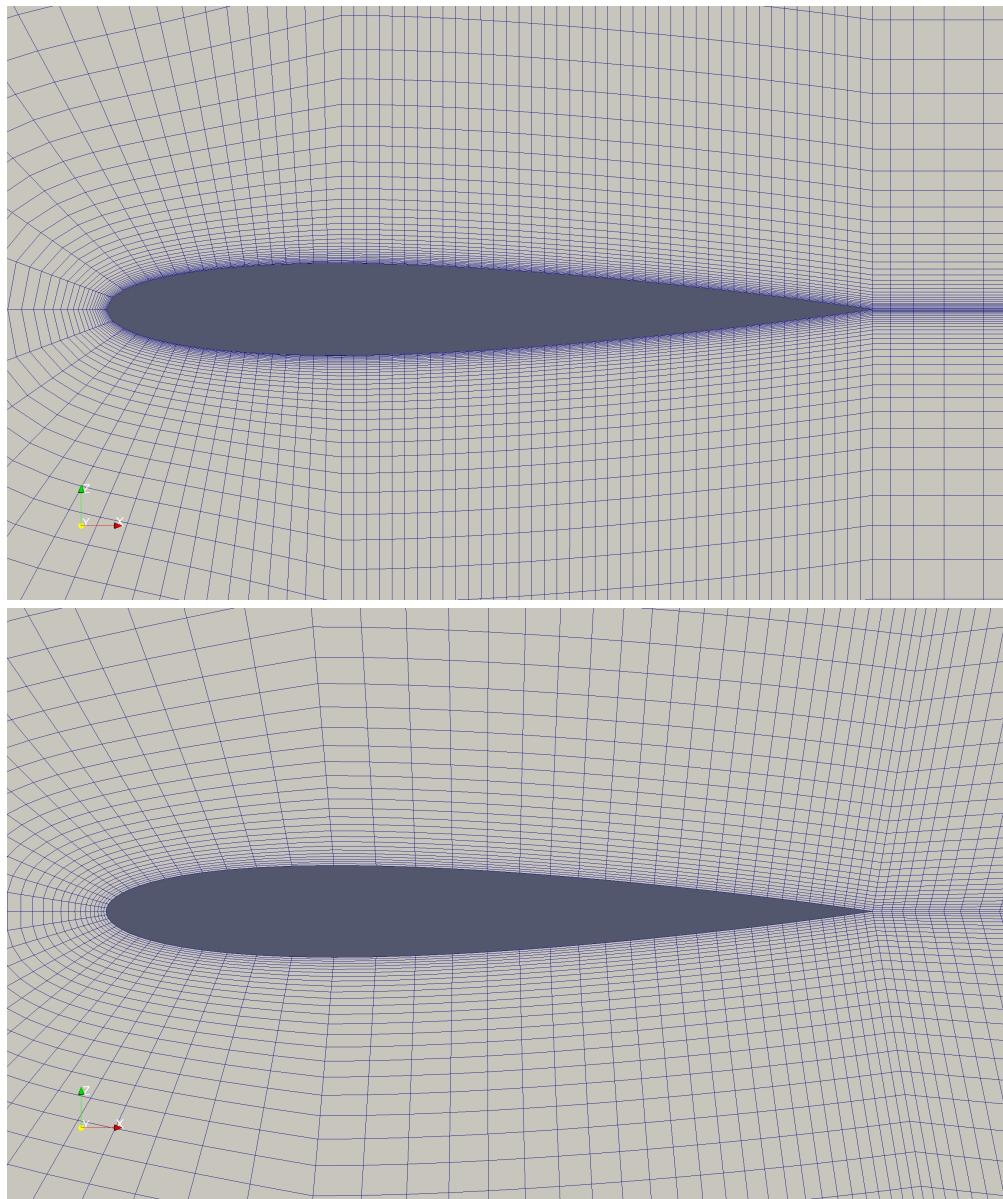
The first attempts have been made using the *meshgen* Matlab/Octave script provided by [19]. This script is able generate a *blockMeshDict* file which is then fed to *blockMesh* for the actual generation of a structured mesh in the *polyMesh* OF format. The code is capable to generate C meshes around any NACA 4-digits airfoils, leaving the freedom to set a number of parameters for the definition of grid's dimensions, resolution and basic refinement.

With the objective of producing a mesh capable to match adjoint's requirements, the original code has been customized, adding the following features:

- Generalized geometry input in the format of plain coordinates file.
- LE and TE refinement.
- Usage of *edgeGrading* blockMesh command instead of *simpleGrading* for better control of refinement.

- Embedded direct and inverse grading calculators (*grading* and *gradinv* functions) to ensure automatic spacing continuity between mesh blocks and for the direct setting of desired cell dimensions on the wall and/or outer boundary.

A comparison of two meshes obtained respectively with the original and customized *meshgen* is provided in fig. 4.2. The latter one has been used extensively in this work in its 300K cells versions, then substituted by meshes generated with *SCMG*.



**Figure 4.2:** Original *meshgen* vs modified: down-scaled 8K cells meshes. The same number of points on airfoil is kept for comparison

### 4.1.3 SmoothCMeshGenerator (SCMG)

The need for a mesh design with smooth transitions between regions and with a powerful control of vertices layout led to the choice of implementing from scratch an unstructured mesh generator. A typical choice for the mesh of an airfoil is a “C” topology. It consists in a rectangular mesh folded around the airfoil in such a way that the central part of the inner contour adheres to the airfoil shape, while the remaining two parts are merged together closing the computational domain behind the LE.

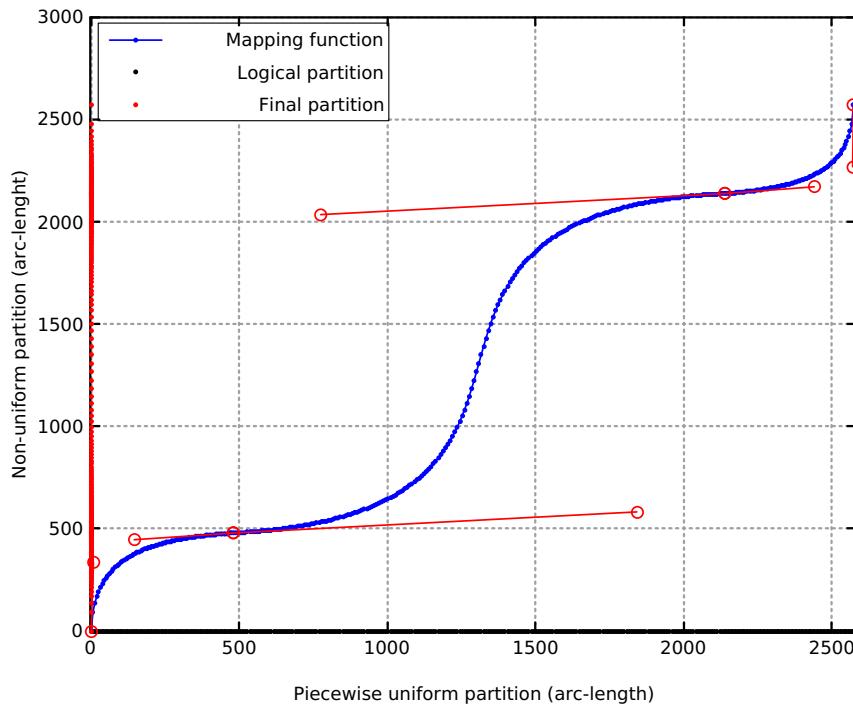
This can be achieved by a suitable mapping of a cartesian  $I \times J$  grid (in the so called “logical space”) into a curvilinear grid (in the so called “physical space”). The idea that motivated this original development was constructing the mapping numerically by defining a few reference contours, complemented with the associated vertex distributions, and by interpolating in between using a family of Bezier curves.

The procedure is briefly summarized in the following steps:

- The body contour is defined through the input of a coordinates file
- Inner and an outer contour (iso-j lines) are defined using piece-wise functions of arc-length, the first one, in particular, is obtained joining the input geometry, which has been interpolated using cubic Hermite splines, with the two coincident straight lines that make up the “wake” contour.
- The vertex layout is then defined on the two curves. This is achieved with a spacing function that maps the “logical” (uniform) partition into the final (non-uniform) one. An handy mechanism for building those function has been found in linking cubic Bezier curves (like in a B-spline), whose “handles” can be used as refinement parameters. Moreover the capability of imposing the slope of start and end points of each Bezier has been exploited to impose the continuity of spacing and to select the desired cell lengths. A visual explanation is given in fig. 4.3.
- The next step is defining the family of iso-i lines, that will connect inner and outer contour in order to complete the “net”. Again a suitable option has been found in cubic Beziers. This time a whole family of curves is needed, for this reason a family of “handles” or control point’s couples must be specified. The idea is to define the locus of control points with the same procedure used to define inner and outer contours. Two additional intermediate contours are

so generated. They act in the same way of 2D “handles” i.e. influencing the way each Bezier transits from one extremity to the other. The concept is pictured in fig. 4.4. A suitable setting allows to obtain lines which are locally normal to both wall and outer contours.

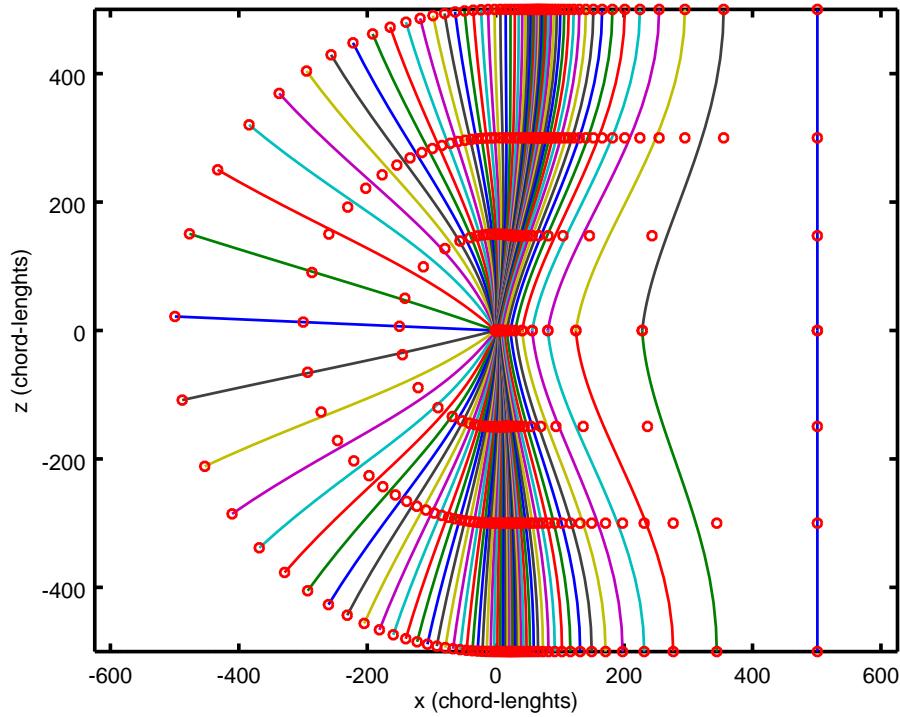
- Finally once the shape of the “net” is ready the last step is shifting the vertices on iso-i lines so to provide a desired “radial” refinement.



**Figure 4.3:** Vertices layout on outer contour: example

## Features

- Generalized geometry input in the format of plain coordinates file.
- Selection of mesh shape and dimensions.
- Advanced refinement settings based piece-wise Beziers definition: control over spacing continuity and cell dimension.
- Orthogonality constraint in cells adjacent to wall and outer boundary, with the possibility of setting the radius of influence of boundary geometry on inner mesh. A smoothing strategy of boundary normals has been introduced

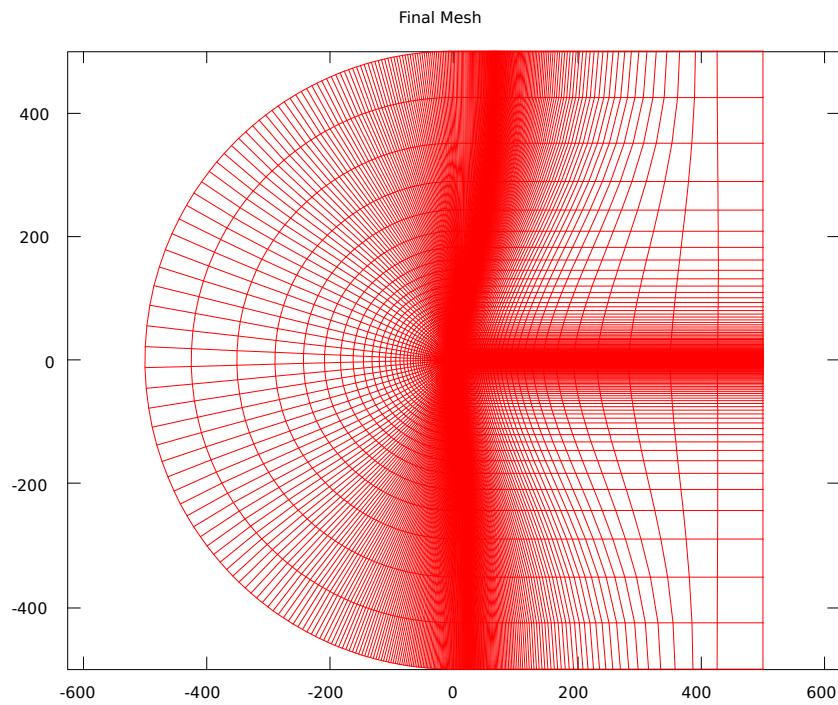


**Figure 4.4:** Vertices layout on inner domain: sampling every 10 lines

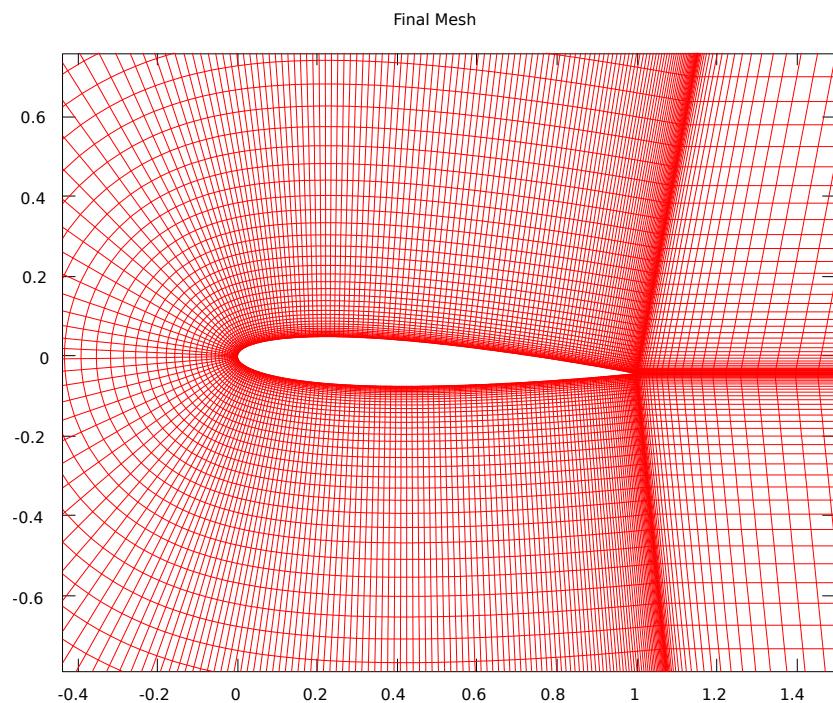
to improve robustness to irregular geometries and locally concave airfoil contours.

- Proportion-preserving scaling of mesh resolution: useful in convergence study and for the creation of mesh templates.
- Mesh crop: selection of a subset of the grid.
- Export to P3D structured mesh format.
- Automated execution of *plot3dToFoam* and *autoPatch* OF utilities for mesh conversion to the OF format and labeling of patches.
- Optional execution of *checkMesh* and *renumberMesh* OF utilities for mesh quality assessment and reduction of matrix bandwidth.

A close and far look to a mesh built with *SCMG* is displayed in figures 4.5, 4.6. This particular mesh mimics the  $897 \times 257$  grid by NASA [6].



**Figure 4.5:** Mesh for NACA 0012 at 2.5 AoA generated with SCMG - far view



**Figure 4.6:** Mesh for NACA 0012 at 2.5 AoA generated with SCMG - near view

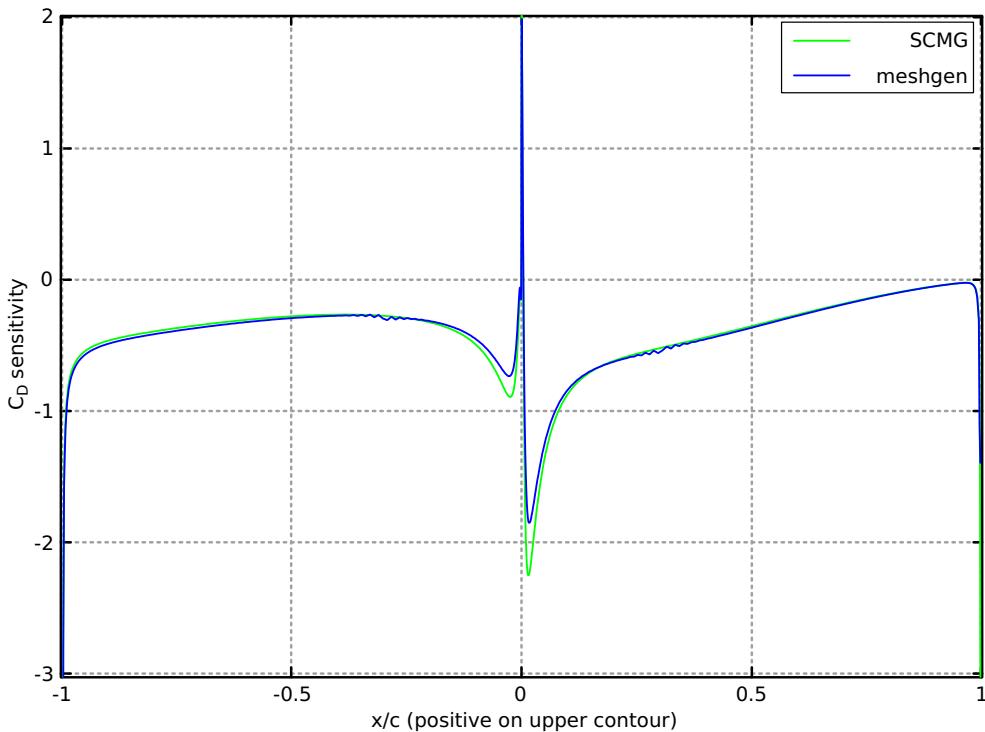
## 4.2 Sensitivity dependence on mesh quality

As expected sensitivity map is conditioned by mesh quality. It is significant to show the comparison of the results obtained on Case B using the two mesh types used throughout the work:

***meshgen v2***: 329K cells - 16 chord lengths radius - 496 points on the airfoil.

***SCMG***: 229K cells - 16 chord lengths radius - 512 points on the airfoil.

Figure 4.7 shows how for the first mesh minor regularity implies a noisy and less accurate solution. In particular, the defect represented by an imperfect conjunction of two blocks at about 30% of the chord, visible in the lower picture of fig. 4.2, results in visible oscillations at two equidistant points from TE. The second mesh design removed the issue and was found to allow a more accurate solution, for this reason it became a standard in this work.



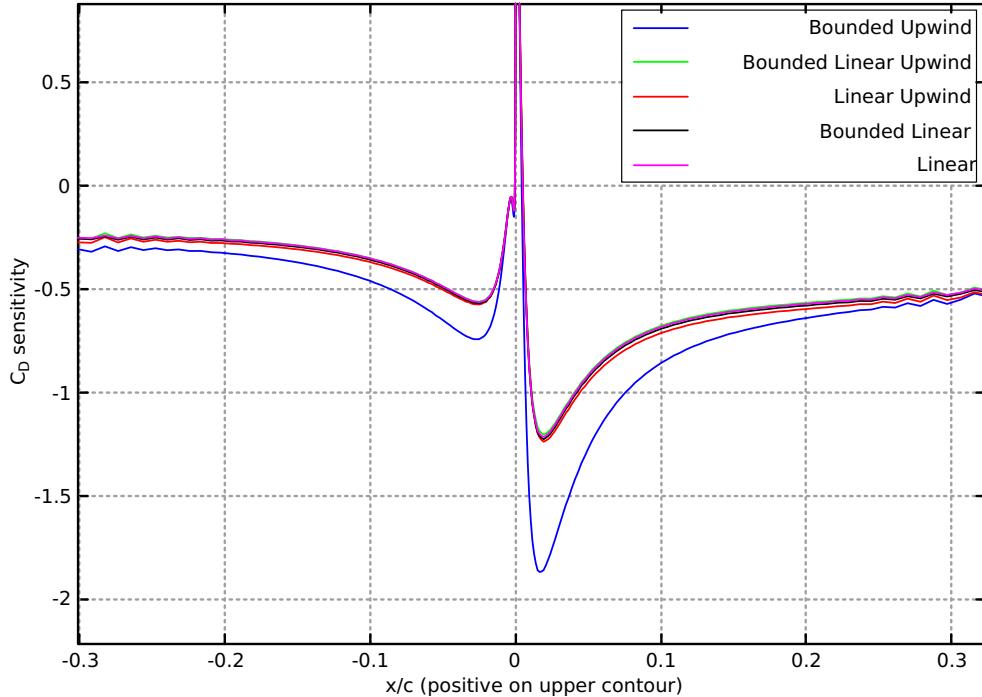
**Figure 4.7:** Comparison of meshes - Case B drag sensitivity computed with TFF<sub>LFA</sub>

## 4.3 Sensitivity dependence on numerical schemes

An important dependence on the scheme used to approximate the advective term of adjoint equations has been found. A comparison based on airfoil drag

sensitivity computed with  $\text{TFF}_{\text{LFA}}$  is displayed in fig. 4.8. It can be noticed that near airfoil's LE sensitivity differs remarkably for *bounded upwind*, which is first order accurate, while the other schemes, which are second order, show almost perfect overlap.

The upwind schemes is acknowledged to be inaccurate due to its high numerical diffusion, yet it's not unusual to find it in turbulence models and in adjoint solvers: it's the case of *adjointShapeOptimizationFoam* and *adjointOptFoam*. Concerning second order schemes: *linear* and *bounded linear* are slightly more accurate if compared to *bounded linear upwind* and *linear upwind*, yet the latter have better stability properties. This aspect has indeed a practical impact: in order to achieve convergence with *linear* schemes under-relaxation of adjoint momentum equation needs to be incremented, leading to an increase of computation time by a factor of 2 or 3. Other more performing measures are thinkable. Finally due to this advantage *bounded linear upwind* scheme has been extensively used throughout this work.



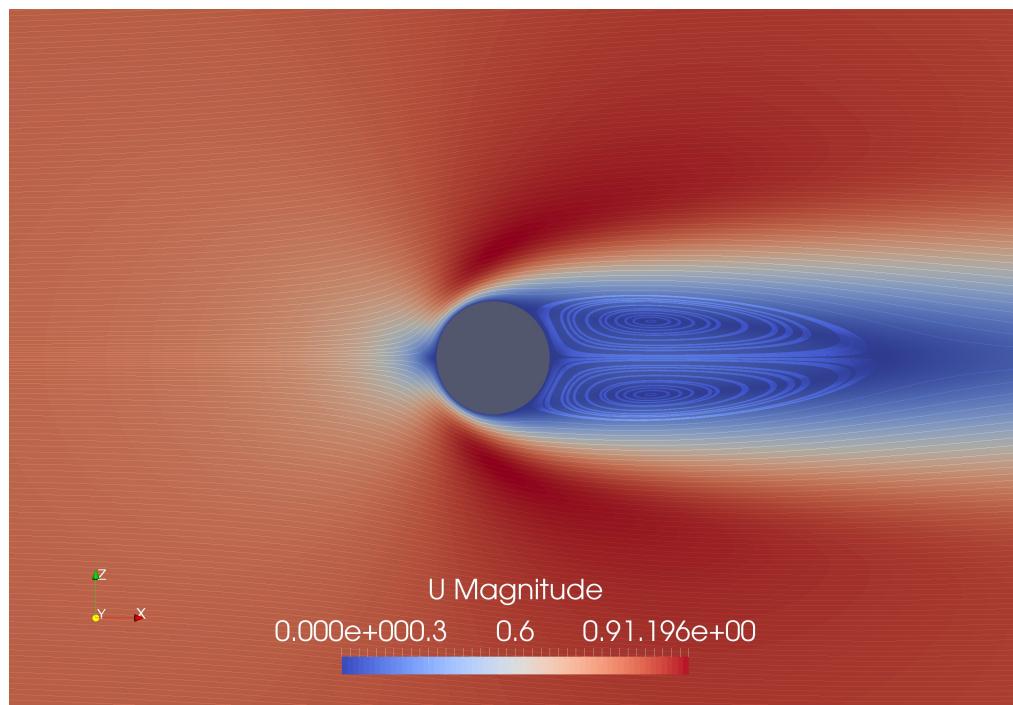
**Figure 4.8:** Comparison of numerical schemes - Case B drag sensitivity computed with  $\text{TFF}_{\text{LFA}}$

## 4.4 Test case A: Cylinder

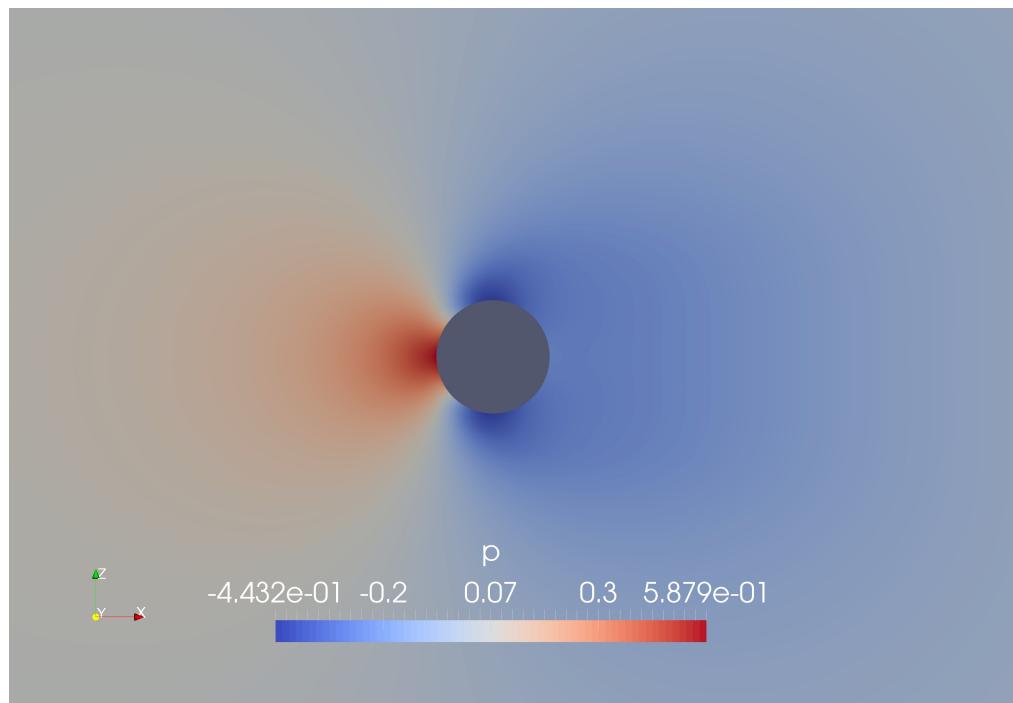
This is a low Reynolds flow condition that features a steady solution. The cylinder represents a blunt obstacle to the flow, that at one point detaches giving raise to a couple of stable counter-rotating vortices, as well known in literature [20]. The flow field and next the dual fields resulting from the three adjoint solvers are shown in figures 4.9a to 4.12b for the case of drag sensitivity.

It can be noticed that adjoint velocity field looks similar for PFF and TFF<sub>LFA</sub> while is nearly the “negative” for MBF. The explanation is found in the different set of BCs: while for the earlier two the body is the motor of the “adjoint flow”, with its imposed adjoint velocity, and the inlet constitutes an impenetrable wall, for the latter the wall is a non slip boundary and outlet and inlet sustain the adjoint flow. Common to all is the formation of an upwind wake, as expected this is due to the upwind transport performed by velocity on its adjoint counterpart. Because of this one way coupling the physical wake affects the adjoint velocity field too, giving rise to what resembles a second wake in the opposite direction. Experience showed that is advisable to grant a proper mesh refinement also in the upwind region.

The inspection of the streamlines of adjoint velocity, here not shown, reveals the presence of complex patterns, including boundary layers, free shear layers and big scale recirculation. While resolving this features can have little importance, knowing the information of interest, i.e. sensitivity, is sampled from the body adjacent cells, the same cannot be told about stability and convergence of the numerical solution. An interesting example is the following: if primal velocity is imposed on the outer boundary, like in the case of a far field BC, or simply on the inlet, the consequent no slip condition on adjoint velocity determines the formation of a boundary layer, which occurs in the region that is usually less refined. This can lead to mesh Péclet based oscillations and to instability or convergence issues. In this regard the usage of symmetry boundary condition has proved to be effective in removing the issue from the respective portion of boundary.

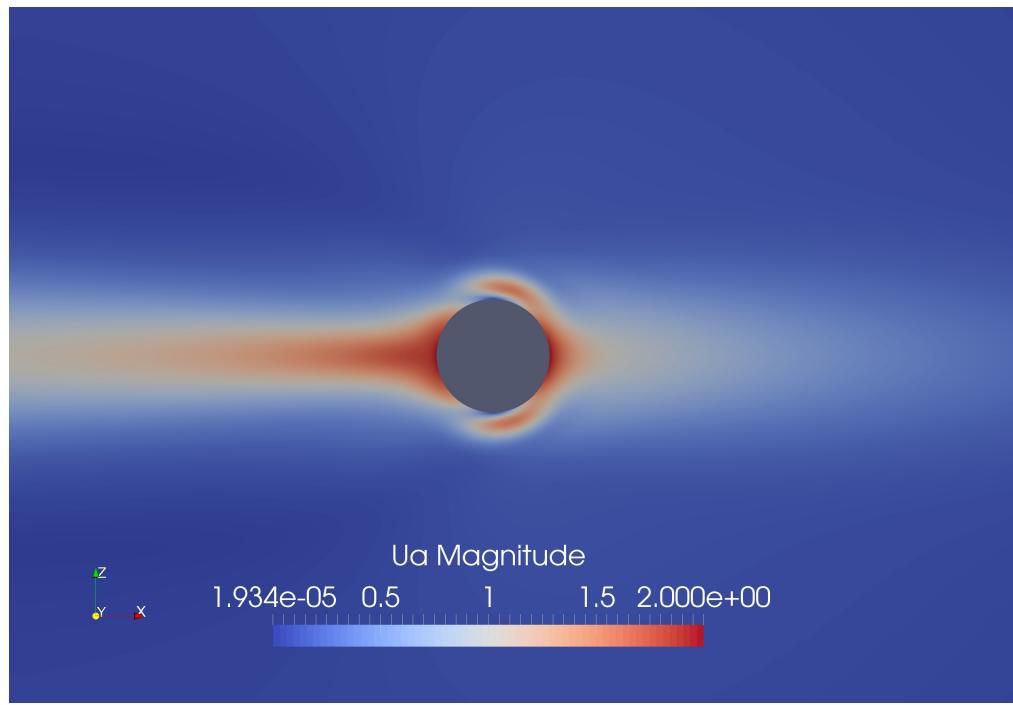


(a) Velocity magnitude and streamlines

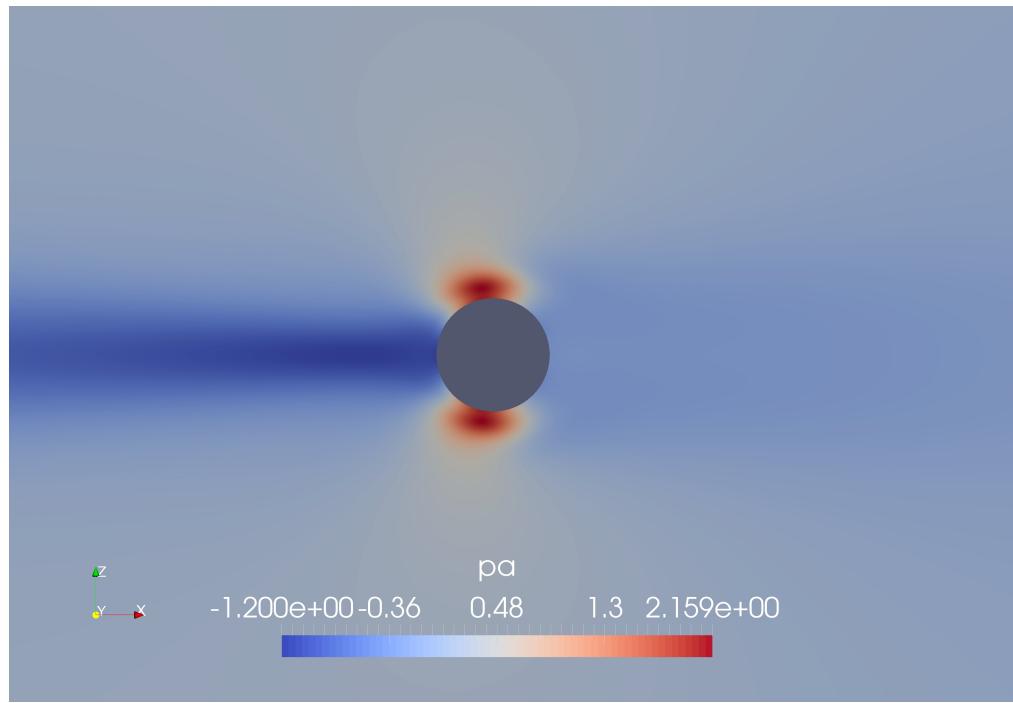


(b) Pressure field

**Figure 4.9:** Case A - Primal fields

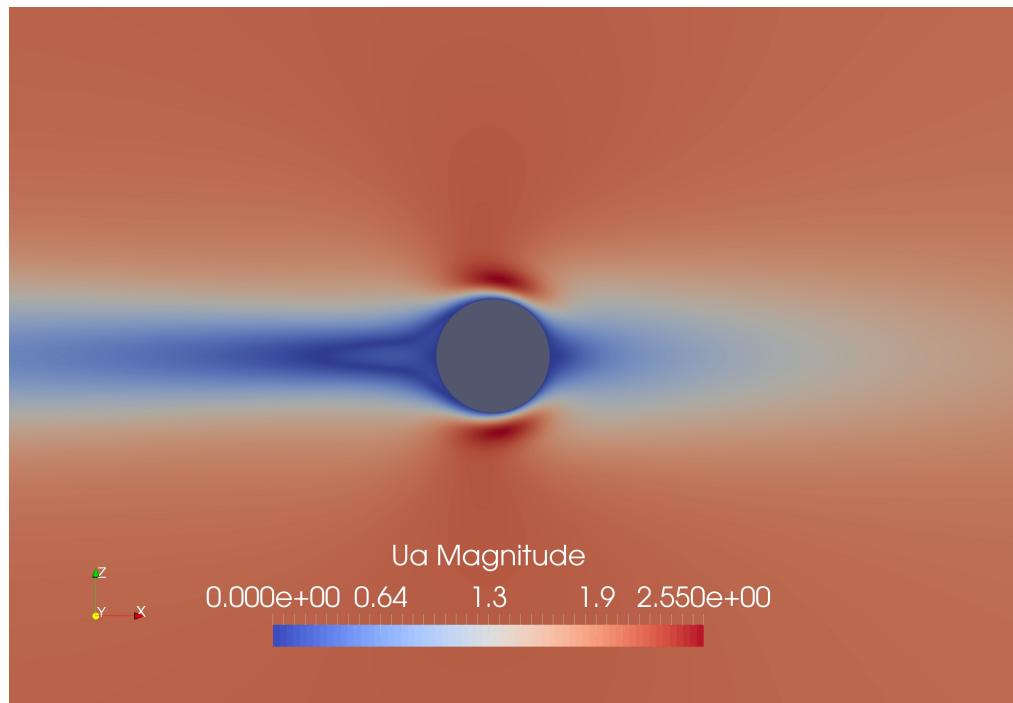


(a) Pressure Force Formulation: adjoint velocity magnitude

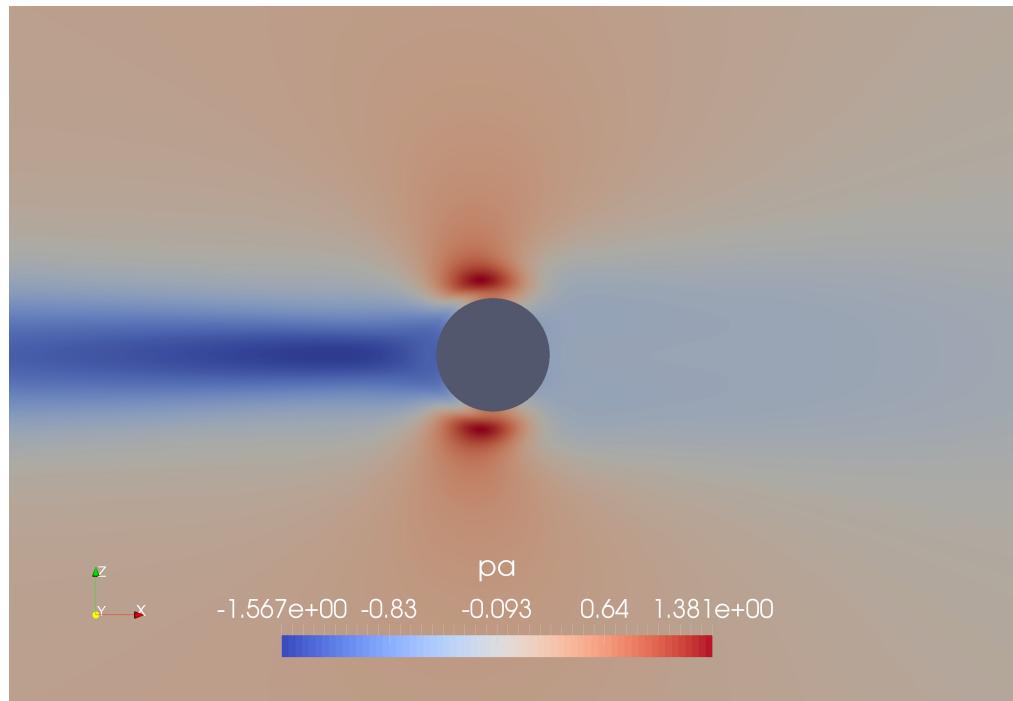


(b) Pressure Force Formulation: adjoint pressure

**Figure 4.10:** Case A - Pressure Force Formulation: adjoint fields

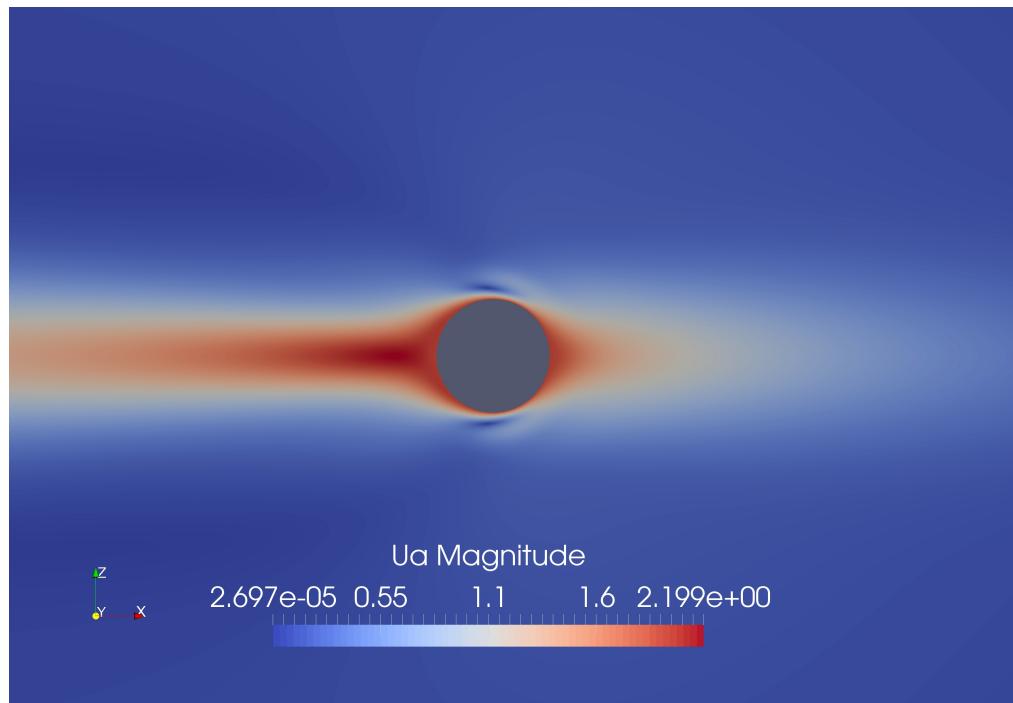


(a) Momentum Balance Formulation: adjoint velocity magnitude

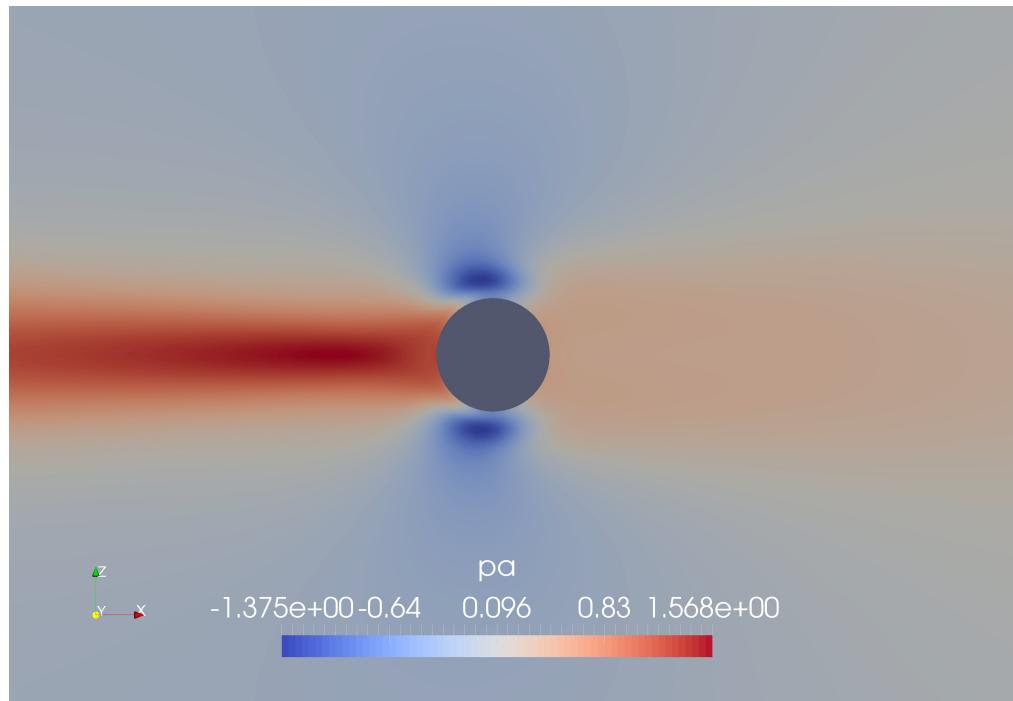


(b) Momentum Balance Formulation: adjoint pressure

**Figure 4.11:** Case A - Momentum Balance Formulation: adjoint fields



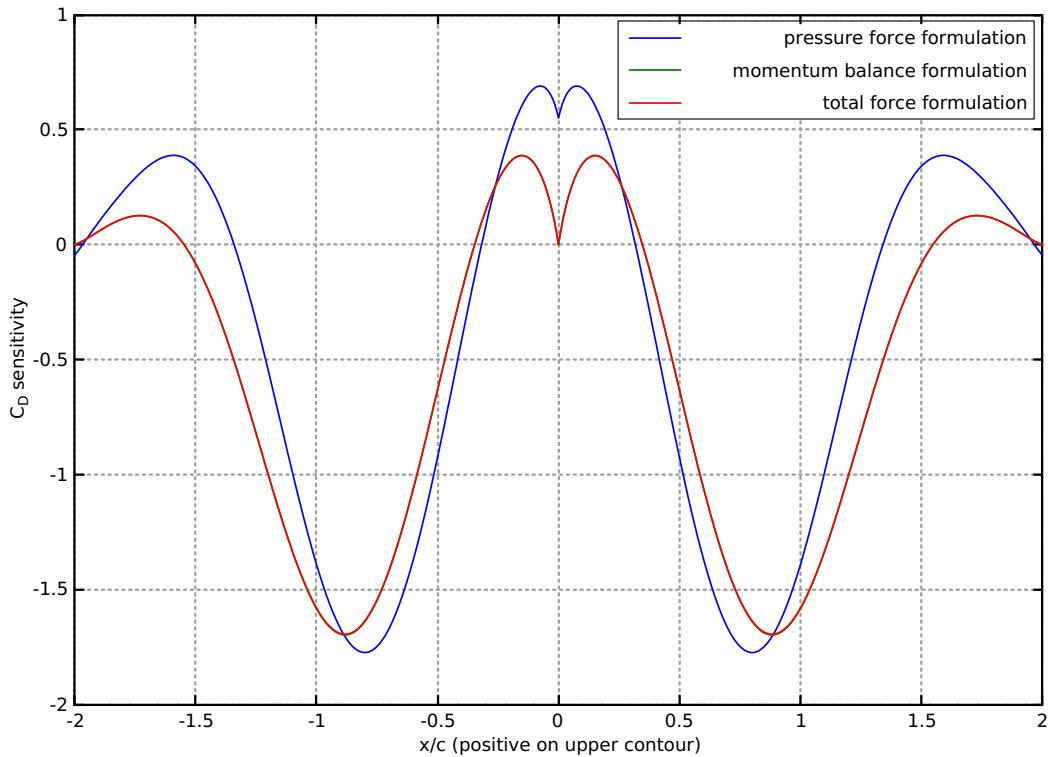
(a) Total Force Formulation - LFA: adjoint velocity magnitude



(b) Total Force Formulation - LFA: adjoint pressure

**Figure 4.12:** Case A - Total Force Formulation: adjoint fields

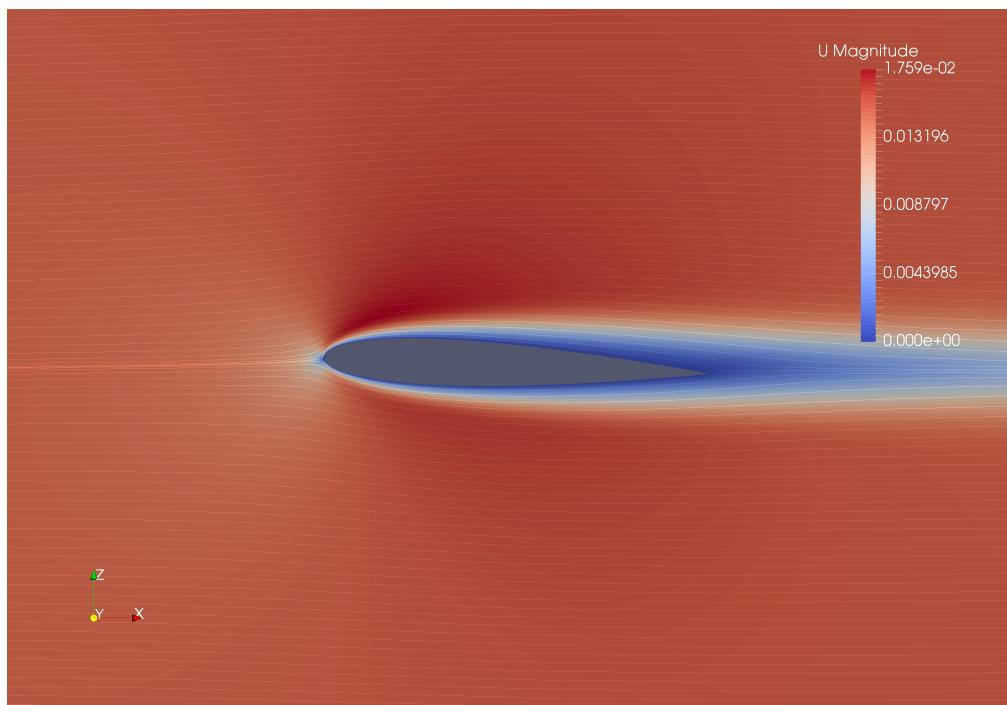
Finally a comparison of drag sensitivity is given in fig. 4.13. The curves are perfectly matching for TFF<sub>LFA</sub> and MBF, while PFF shows peaks of different position and magnitude and in general sharper traits. The diverse shape of pressure force sensitivity can be in general justified by the fact it captures of only a part of involved physics, yet the difference in magnitude is not expected as by definition pressure sensitivity should be a fraction of total force sensitivity. This flaw is likely due to poor discretization of the higher order derivatives, as suggested by figure 3 of [1]. Recalling the convention of normals (positive pointing inward) and considering  $C_d$  is reduced deforming as the opposite of gradient (i.e. a second change in sign), the result can be read as: drag reduction is achieved by compressing equally the cylinder along the vertical axis and by stretching it in the horizontal direction, especially in the upwind side. A curious and aspect is that at both front and rear stagnation points sensitivity is zero, encouraging the creation of two couples of protrusions.



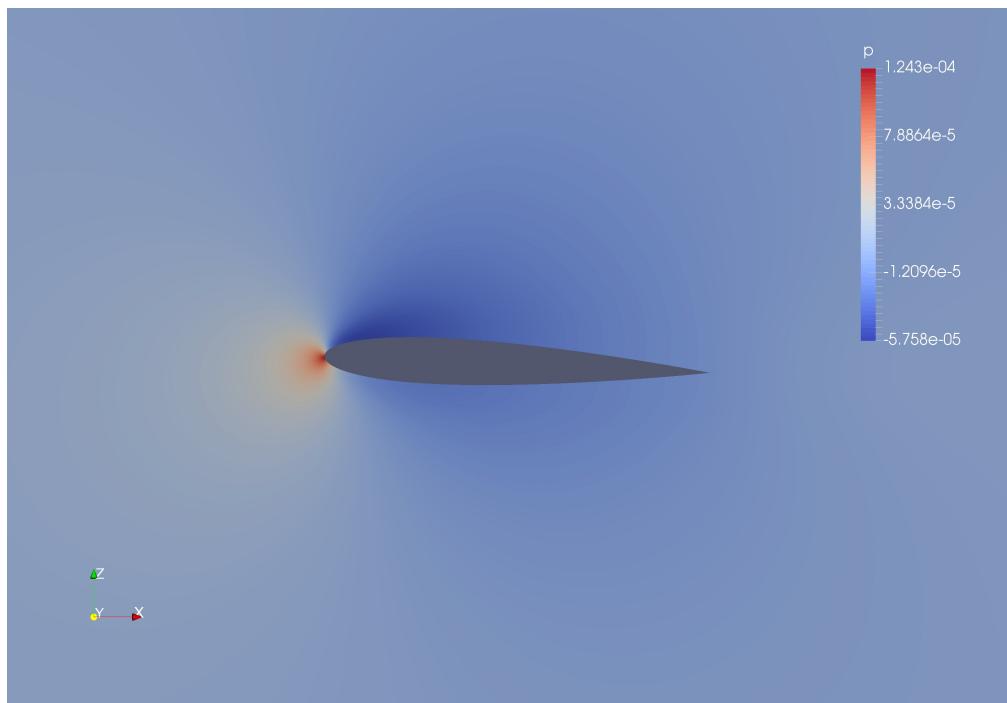
**Figure 4.13:** Case A - Drag sensitivity comparison (green and red curves overlap)

## 4.5 Test case B: Airfoil

This configuration features a fully attached and laminar flow. Primal and adjoint flows are shown in figures from 4.14a to 4.17b. The considered cost function is drag coefficient. The fields share a similar appearance with the cylinder case. No recirculation bubbles are present in the primal field, but the presence of a lifting body poses a different physical condition. The features of “adjoint flow”, mentioned in the previous case, became more convoluted and small due to a higher Re number.

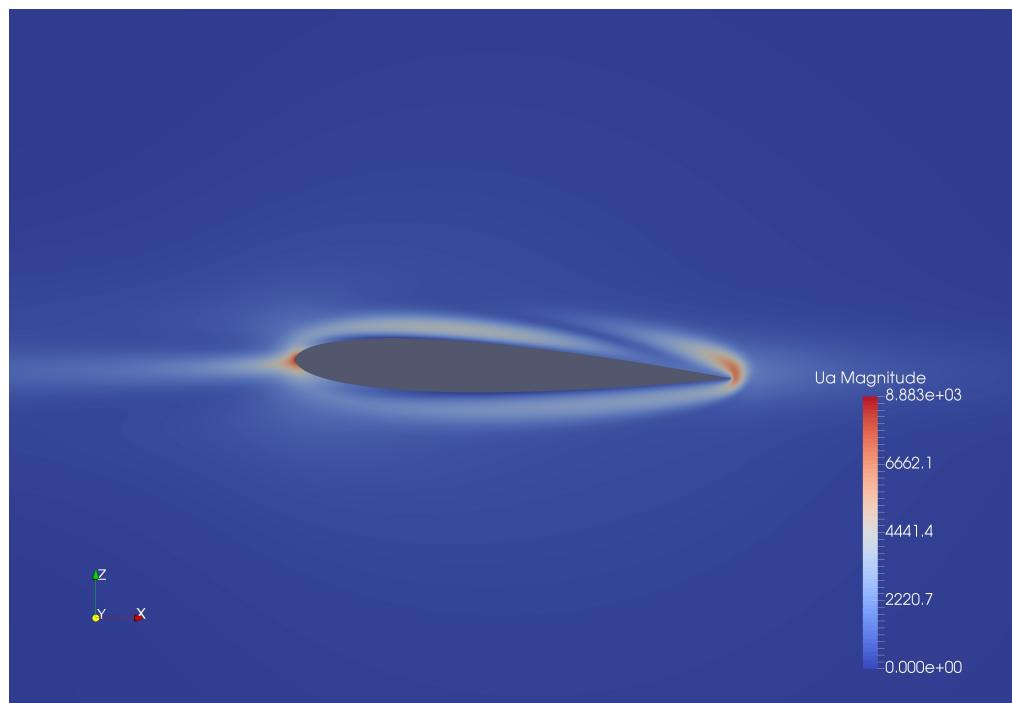


(a) Velocity magnitude and streamlines

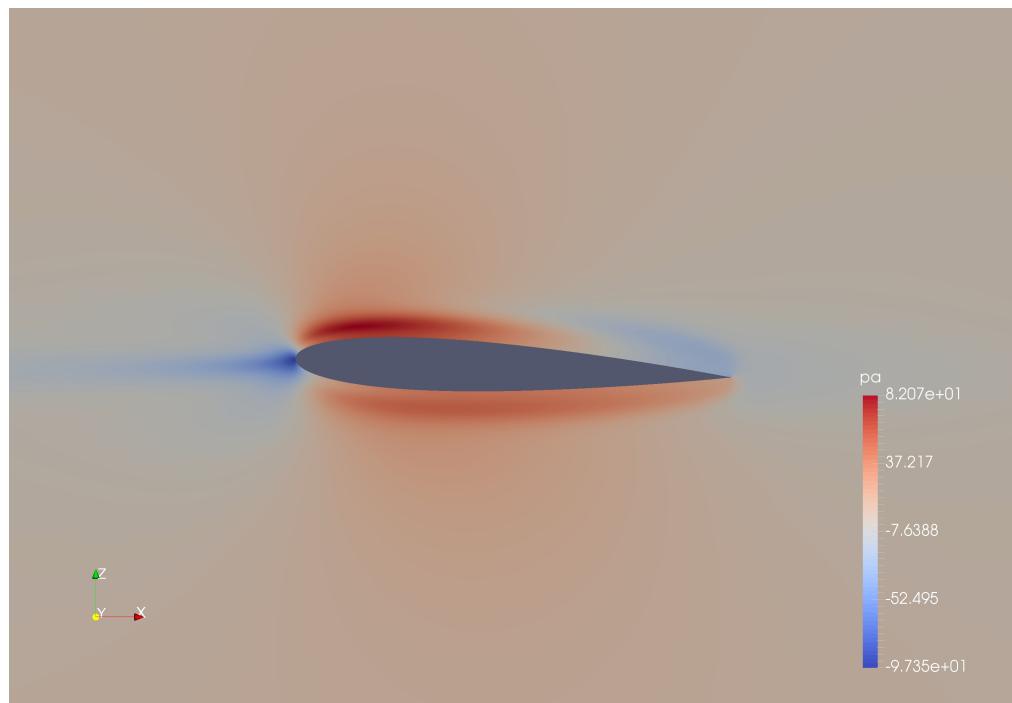


(b) Pressure field

**Figure 4.14:** Case B - Primal fields

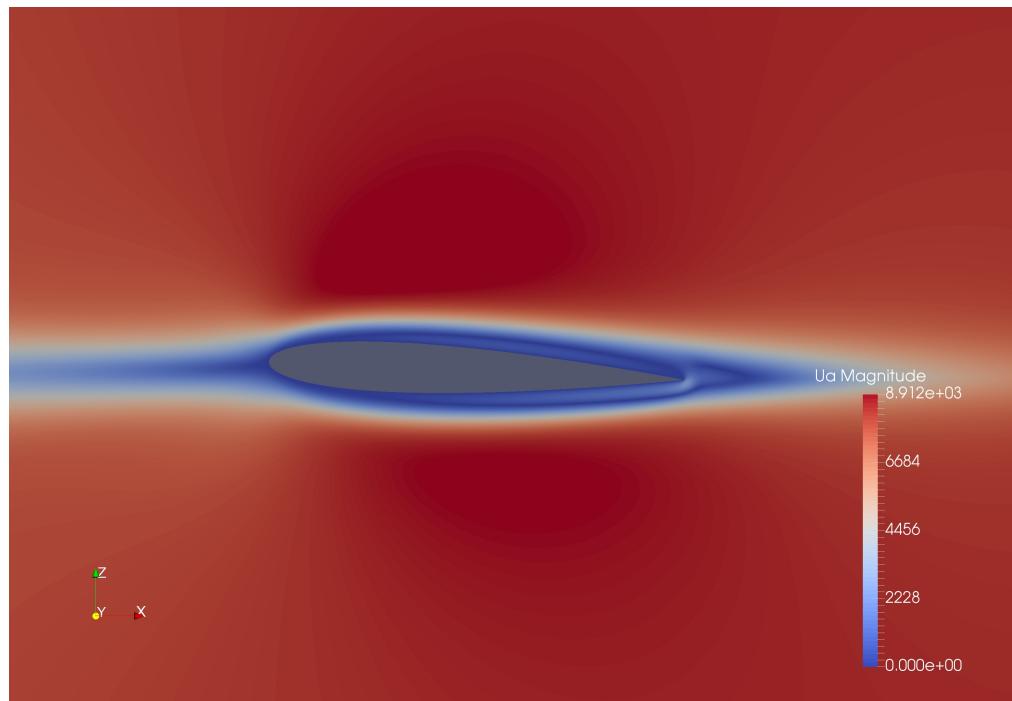


(a) Pressure Force Formulation: adjoint velocity magnitude

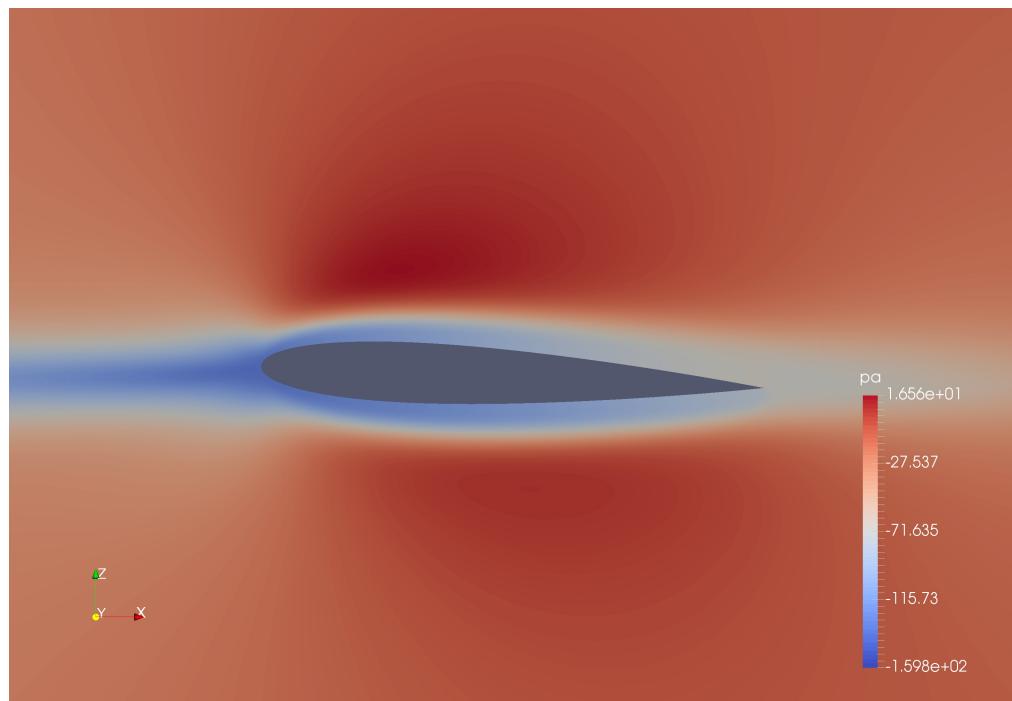


(b) Pressure Force Formulation: adjoint pressure

**Figure 4.15:** Case B - Pressure Force Formulation: adjoint fields

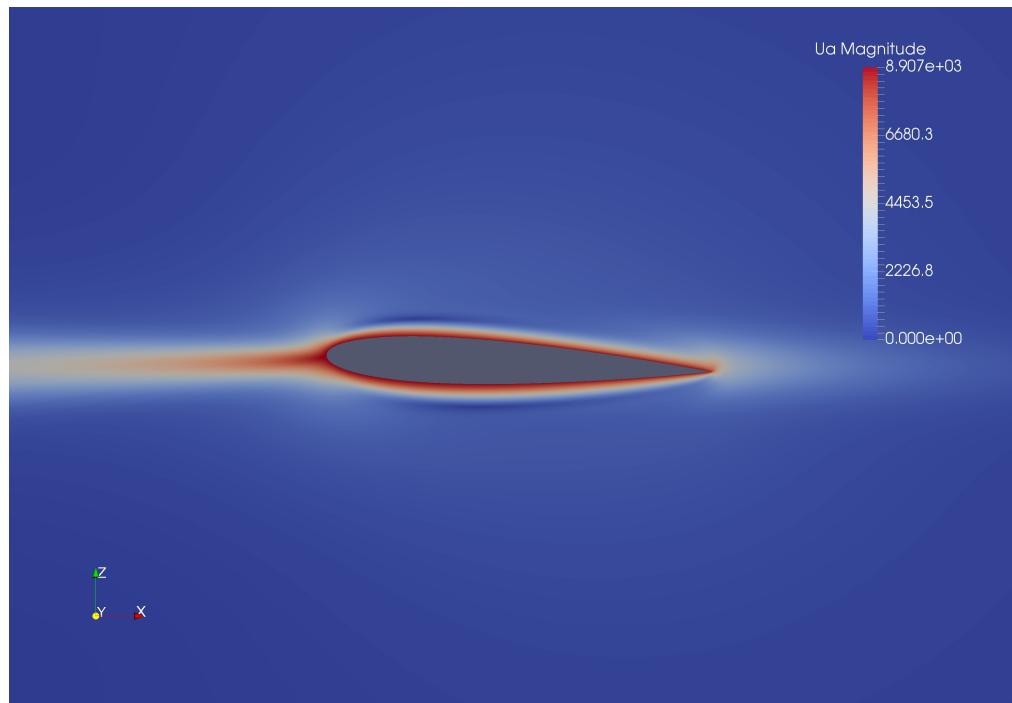


(a) Momentum Balance Formulation: adjoint velocity magnitude

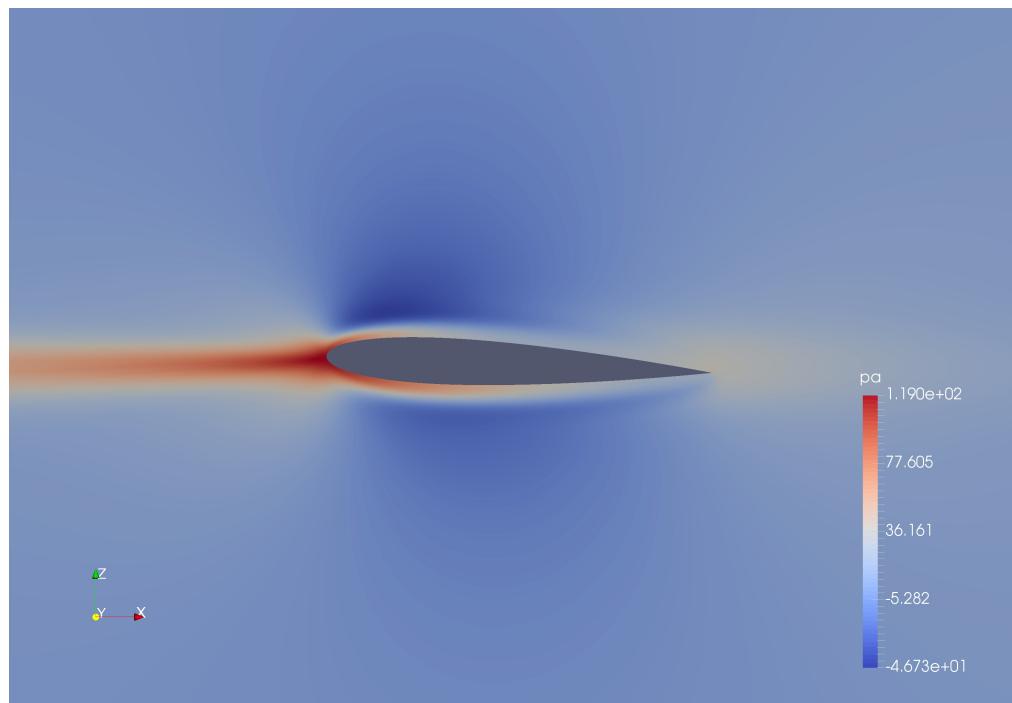


(b) Momentum Balance Formulation: adjoint pressure

**Figure 4.16:** Case B - Momentum Balance Formulation: adjoint fields



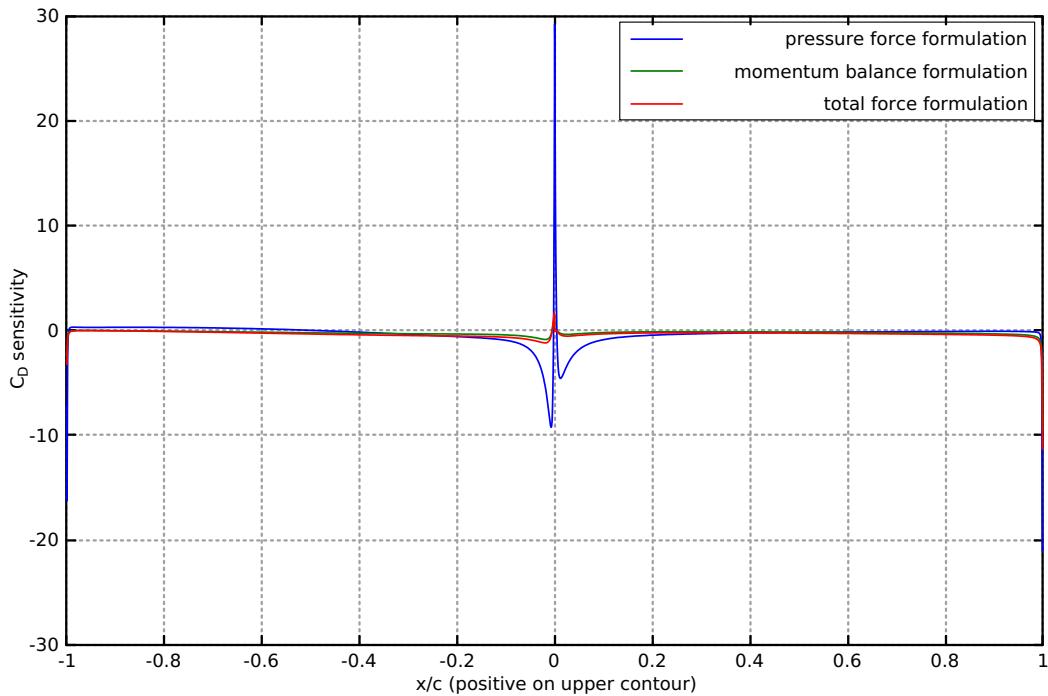
(a) Total force formulation: adjoint velocity magnitude



(b) Total Force Formulation: adjoint pressure

**Figure 4.17:** Case B - Total Force Formulation: adjoint fields

Finally a comparison of drag sensitivity is given in fig. 4.19. A similar situation than in case (A) is found: the blue curve (PFF) differs remarkably from the others, while the other two show a discrete accord. The difference in magnitude and shape of pressure force sensitivity seems to be emphasized by the higher Reynolds of this test case. However a common pattern is evident, showing a similar optimization strategy is suggested.



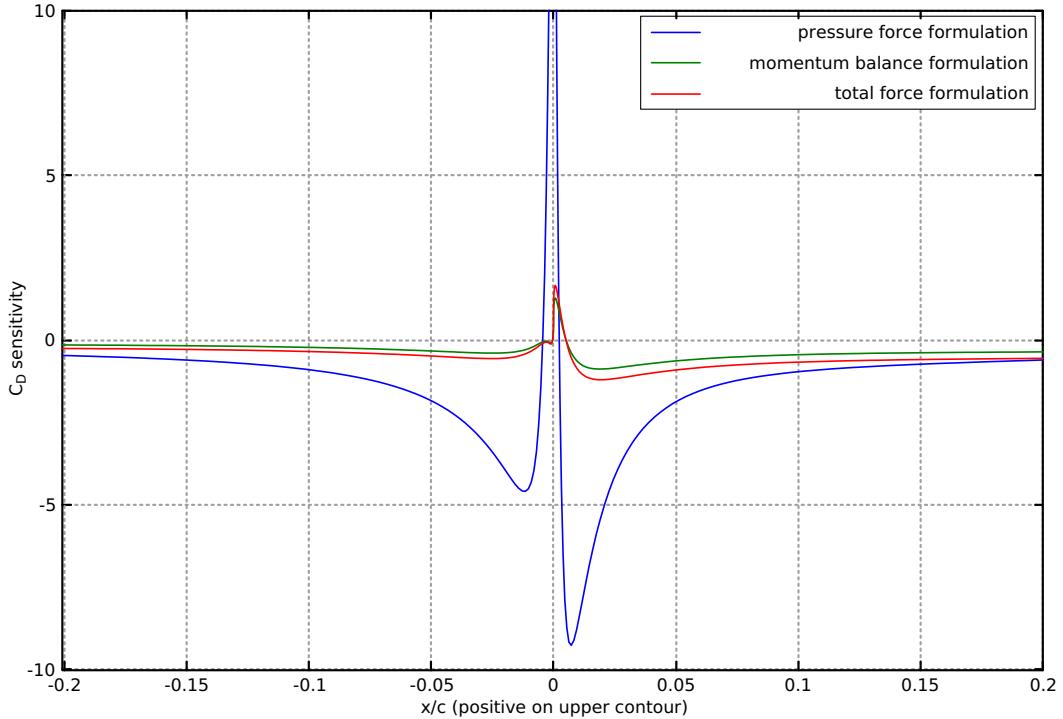
**Figure 4.18:** Case B - Drag sensitivity comparison

## 4.6 Discussion of methods

Follows a review of the variants based on theoretical aspects and on the experience gained with test cases.

### 4.6.1 Pressure Force Formulation

The definition of a cost function based only on pressure translates into a simpler evaluation of sensitivity, however not higher-derivative-free. The restriction to the effect of pressure alone represents a loss in information, which can be considerable in the case of drag sensitivity of streamlined bodies or wings, while in the latter case lift sensitivity is obviously less affected.



**Figure 4.19:** Case B - Drag sensitivity comparison: magnified view

Despite its theoretical limits, this formulation proved to be effective in constrained optimization of airfoils, both in laminar [3] and turbulent conditions [4], leading to an increase of lift from 15% to 100% and to a drag reduction from 1% to 9%. This shows that while performing better in optimizing lift, a drag reduction is also possible. The solver is expected to achieve its best results in the case of turbulent flows around blunt bodies, in which most of drag is due to pressure drag. Indeed the solver has been applied to a one step optimization of a car design in [4], gaining a drag reduction of around 4% with a single small deformation opposed to drag sensitivity map. Finally the presence of higher order derivatives in the expression of sensitivity makes it subject to noise, which is emphasized by any mesh defect or irregularity.

### 4.6.2 Momentum Balance Formulation

This alternative formulation takes advantage of lesser regularity requirements and richer sensitivity: the former aspect decreases the amount of noise in the sensitivity map and the latter gives it new features and details. A drawback is posed by the likely degradation of accuracy due to a poor mesh resolution in the wake region and, as mentioned earlier, a greater dependency on the degree of

convergence of the primal equations. Deep convergence of residuals can be an unfavorable requirement in industrial applications, as robustness of results can be affected. This topic however has not been systematically investigated in the present work.

Another aspect is that the indirect definition of the cost function it's unfavourable when the portion of interest is only a part of the body or in the case of multiple bodies. The point is that due to non linearity it's almost impossible to separate the effect of different parts on the far wake, yet this might not be important as we might restrict only control to a given subset of body surface and keep a cost function based on the overall drag. This weakens the generality of the method, but not that much.

### 4.6.3 Total Force Formulation - LFA

This formulation features the same benefits of the momentum balance version, yet not the same drawbacks. Thanks to the local definition of the cost function, sensitivity appears to be mostly dependent on the near-body primal and adjoint fields, this results in looser requirements on far mesh resolution. Even in the case of numerical oscillations due to high mesh Peclet in the vicinity of the inlet, sensitivity is mostly unaffected.

For the same reason the extension of the setup to multi-body configurations or to the evaluation of specific part of a design is straightforward, making the method capable of the desired generality. These qualities are welcome in the perspective of the practical applications.

# Chapter 5

## Validation

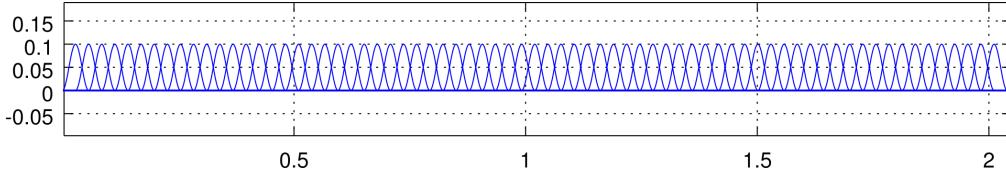
The definitive step in assessing the accuracy of the methods herein presented is validation with finite differences. Keeping the paper of Castro et al. as main reference for comparison, we decided to base validation on the airfoil case (B), which showed to be more challenging.

### 5.1 Parametrization

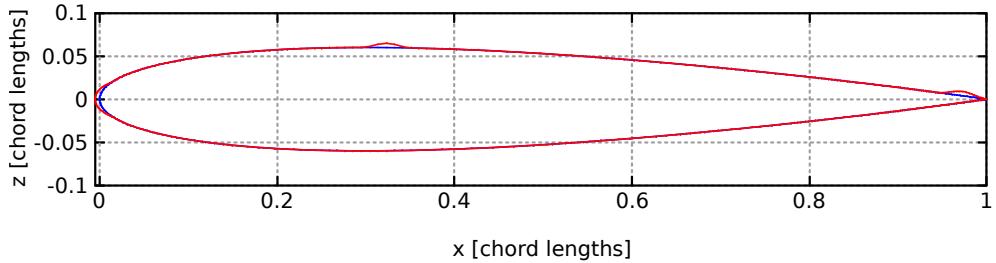
The first choice concerns the design space to which finite differences are restricted, this should be a trade-off between accuracy and computational effort. A viable choice is to use a parametrization for the deformation field, recasting then sensitivity on a smaller set of degrees of freedom instead of performing a costly node by node comparison. In the paper a parametrization based on Hicks-Henne functions was used, which is a common choice for optimization of airfoils. Yet this choice was not considered optimal for our purposes for the reason hereafter explained.

The gradients w.r.t. parameters represent a projection of nodal sensitivity on the reduced design space, which in other words entails a filtering with loss of information. In the case of H.H. functions, whose support is defined on the whole chord, this filtering is intense and the profile of sensitivity is smoothed and spoiled of details.

In order to produce a finer comparison a set of shape functions characterized by compact support has been used, specifically sine-squared bumps continuously and homogeneously distributed on the body surface. Each bump function applies a normal displacement profile in the neighbourhood of its control point, the convention of positive outward deformations has been adopted. The final setup comprises 71



**Figure 5.1:** Set of sine-squared shape functions

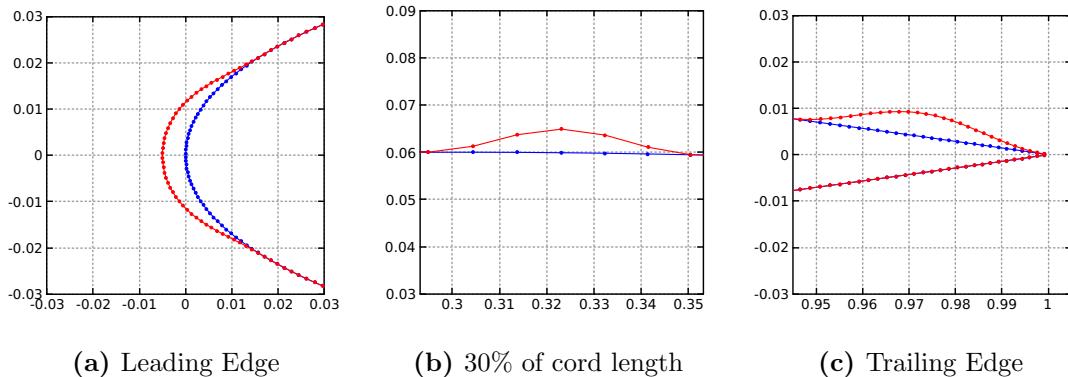


**Figure 5.2:**  $5 \cdot 10^{-3}$  peak displacement in nodes 36, 50, 71

control points equally spaced in arc length and ordered clockwise beginning from TE. A representation is given in figs. 5.1 and 5.2, the latter shows an example of a scattered deformation of magnitude 0.5% of the chord.

## 5.2 Case setup

A structured C mesh of around 200k cells has been used with 512 nodes on the airfoil and an extension of 16 chord lengths in every direction. A code for building smooth C meshes has been developed for the purpose. The setup of the grid has been a compromise between having enough wall resolution to perform surface deformation straightforwardly and ensuring a good deal of uniformity and



**Figure 5.3:** Caption

regularity to favour the adjoint solver. The best consistency of results is obtained in fact using the same grid for both primal and adjoint, however decent results are obtained also using two problem specific grids, which can be a viable option for big scale industrial problems. A core aspect in keeping an high accuracy is the quality of interpolation of the primal field on the usually finer adjoint grid.

Concerning the finite differencing, a sensitivity study has been carried in order to select a proper increment, which is problem and even position dependent. Finally a displacement of  $10^{-4}$  chord lengths for every component has been chosen as good compromise between filtering numerical noise and capturing the linear variation. An important aspect that the study brought to attention is the accuracy in computation of force coefficients. If the same degree of convergence is attained for coefficients, upon subtraction the error due to residual change is erased, if this is not the case, big deviation can be experienced in the gradient. The usual termination criteria based on residuals doesn't imply the condition is respected, for this reason two alternatives are possible: either bringing simulations to deep convergence, as suggested in [21], or using a termination criteria based on coefficients. The latter choice has been pursued, being resource efficient. For the purpose a function object has been implemented in OpenFOAM, and a simple criteria based on averaging of samples has been used. The residuals on coefficients have been selected to be small compared to the typical increment in the cost function. Forward differences have been used, meaning  $n+1$  evaluations of the primal problem are required by one gradient calculation.

## 5.3 Results

The comparison of finite differences and adjoint methods is shown for drag and lift gradients. As expected the gradient w.r.t. bump functions closely resembles nodal sensitivity and the situation portrayed by fig. 4.19 is unchanged in fig. 5.4. The result points out that the drag sensitivity provided by the last two methods is correct, particularly for TFF<sub>LFA</sub>. A different situation is found for lift sensitivity in fig. 5.5: the pressure force variant improves dramatically its performance demonstrating a good behaviour in predicting the variation of lift; the accuracy of Total Force Formulation is confirmed.

The lagrangian-free variant of TFF has proved to be by far the most accurate, showing a precise overlapping with finite differences and no amplification behaviour. The details can be appreciated in figs. 5.6-5.7.

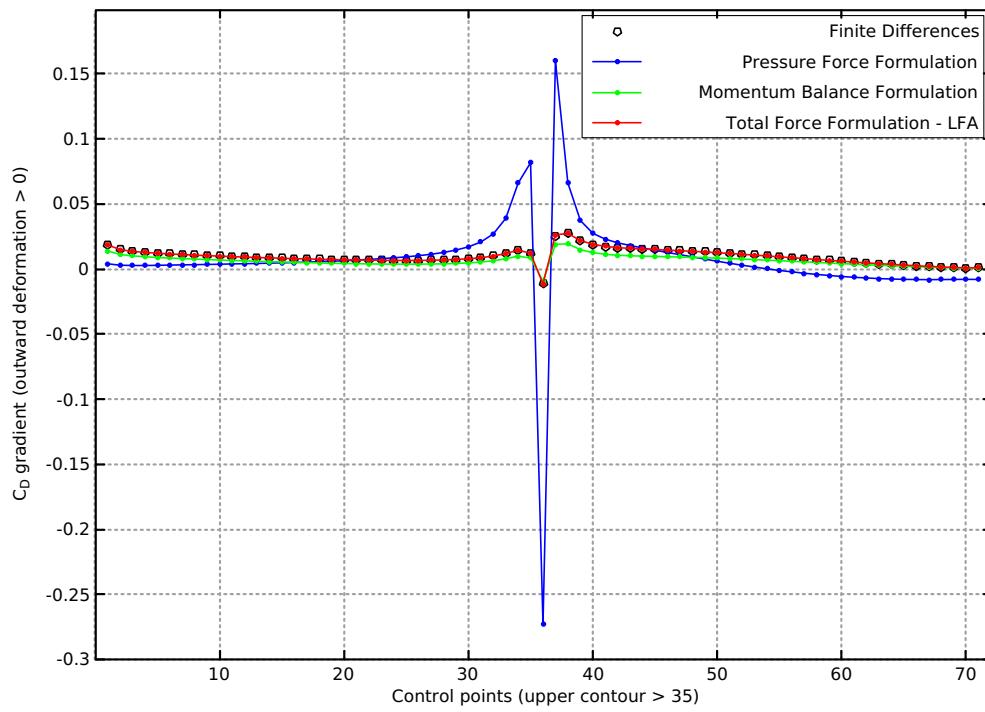


Figure 5.4: Finite differences vs adjoint methods:  $C_D$  gradient

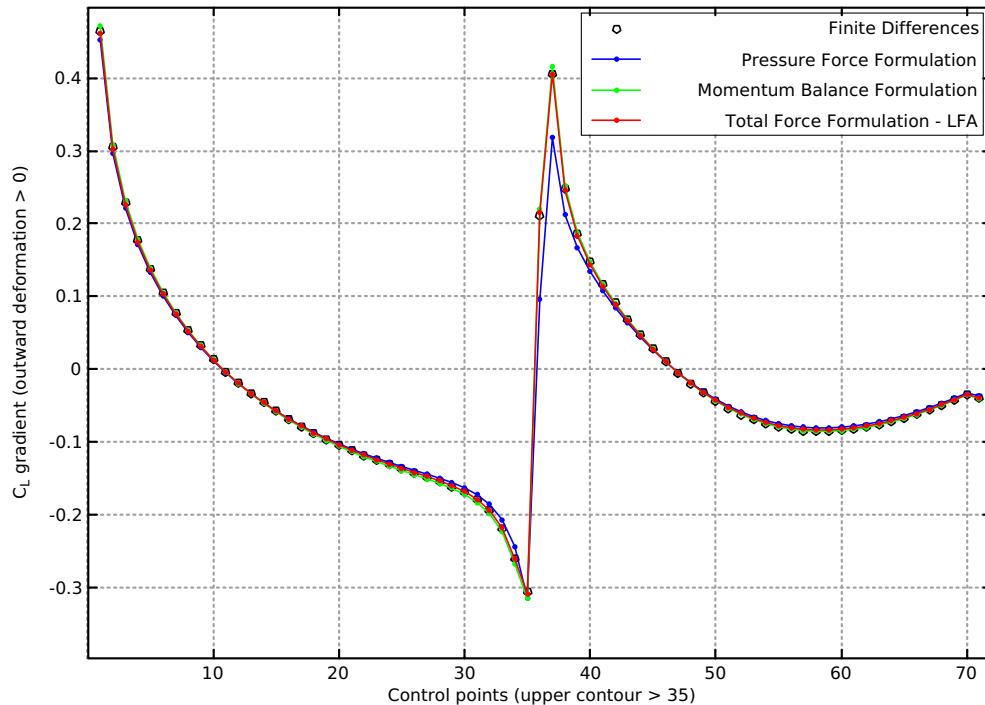
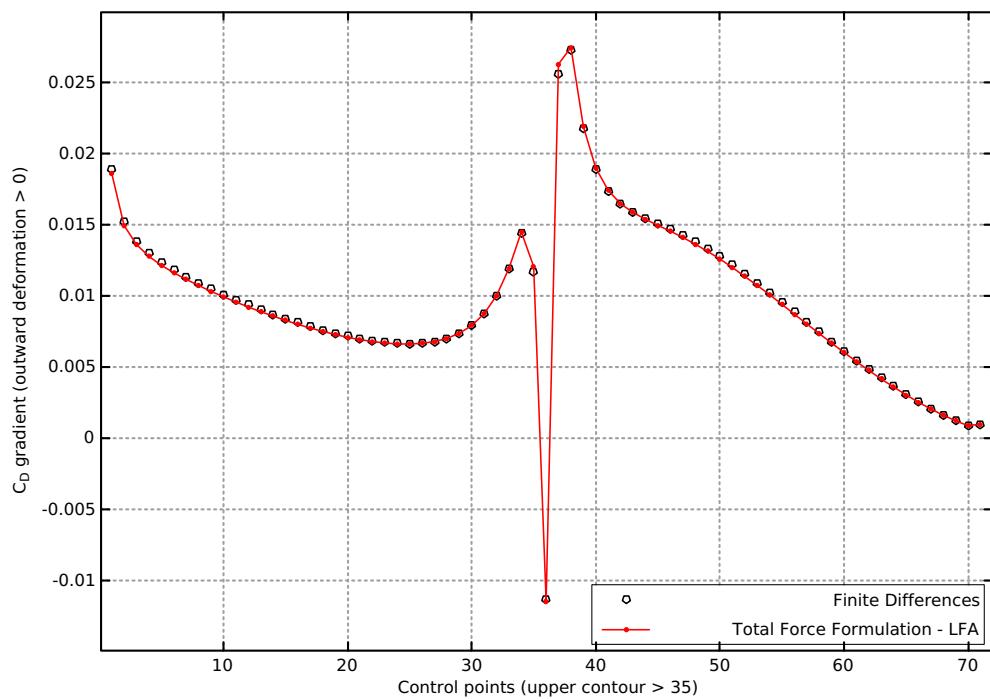
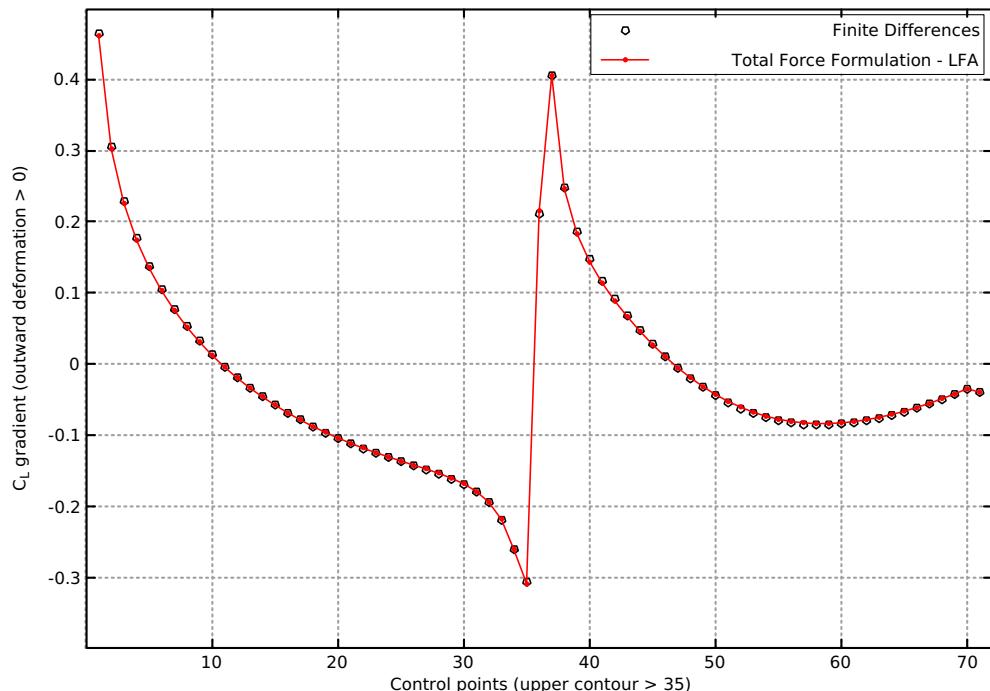


Figure 5.5: Finite differences vs adjoint methods:  $C_L$  gradient



**Figure 5.6:** Finite differences vs Total Force Formulation - LFA:  $C_D$  gradient



**Figure 5.7:** Finite differences vs Total Force Formulation - LFA:  $C_L$  gradient

# Chapter 6

## Conclusions

The present work has addressed the adjoint method for shape optimization in external aerodynamics. We have considered the continuous adjoint formulation, starting from its foundations, aiming at a clear and comprehensive mathematical description. Our efforts have been rewarded by an increased confidence in the results, and by the ability to put the available results into a richer perspective.

The two main theoretical approaches available [5, 8, 1, 9] are first described in general terms, and then specialized to the case of interest. A set of formulations is derived and tested on laminar 2D flows, to highlight potential issues and opportunities, as well as to compare the solution of an identical problem. Specifically, a method including pressure forces only (PFF) is extended to a full force formulation; two variants are described, one based on a novel reformulation of the external aerodynamics problem (MBF), and a second based on the approach of Castro et al. [1] ( $TFF_{LFA}$ ).

Moreover, a procedure for direct evaluation of gradients by finite differences has been applied to the airfoil test case. Comparing the sensitivities to those obtained using the adjoint method consolidates the overall validity of the approaches, and displayed a net improvement in the former two methods over the first one.

The PFF, a revisited version of the method used in [3] and [4], achieved reasonable results concerning the gradient of the lift coefficient, while a severe overestimation of the gradient of the drag coefficient is observed. As suggested in [1], in fact, PFF would require accuracy higher than second-order, owing to the presence higher-order derivatives.

The second method (MBF) was the best option within the LBA class, achieving correct results despite the the fact a completely different cost function is enclosed. This diversity is indeed the cause of an increased susceptibility of gradients to the

order of accuracy and to the magnitude of primal and adjoint equations' residuals.

The third method ( $TFF_{LFA}$ ) yielded excellent agreement with finite differences data, and in addition is characterized by an interesting degree of robustness, so that we consider it as the best of the considered formulations.

It must be emphasized that all three methods show a strong tendency to amplify even tiny grid defects, and a remarkable dependence upon the order of accuracy of the discretization schemes. On the basis of our observations, second-order accuracy seems to be required for satisfactory results.

In conclusion, a satisfactorily accurate, robust and validated methodology for calculating the sensitivity gradients has been obtained. This shows that adjoint optimization can be successful with the finite volumes discretization typical of industrial CFD solvers, provided due care is taken. The present results are encouraging in view of the future extension of the present work to turbulent flows over three-dimensional geometries.

# Bibliography

- [1] C. Castro, C. Lozano, F. Palacios, and E. Zuazua. Systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA journal*, 45(9):2125, 2007.
- [2] R. Pieri. Stanford university unstructured: un software innovativo per la simulazione e l'ottimizzazione geometrica. Master's thesis, Politecnico di Milano, 2013.
- [3] M. Murari. Ottimizzazione vincolata adjoint-based: sviluppo di una metodologia open source su piattaforma HPC. Master's thesis, Politecnico di Milano, 2015.
- [4] L. Palma. Ottimizzazione in correnti turbolente con il metodo delle equazioni aggiunte: sviluppo di una metodologia su piattaforma HPC. Master's thesis, Politecnico di Milano, 2015.
- [5] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *Int. J. Num. Meth. Fluids*, 58:861–877, 2008.
- [6] NASA Langley Research Center. 2D NACA 0012 Airfoil Validation Case: Grids - NACA 0012 Airfoil Case. [https://turbmodels.larc.nasa.gov/naca0012\\_grids.html](https://turbmodels.larc.nasa.gov/naca0012_grids.html), 2015.
- [7] J.E.V. Peter and R.P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010.
- [8] O. Soto and R. Löhner. On the computation of flow sensitivities from boundary integrals. *AIAA Paper*, 112:2004, 2004.
- [9] F. Palacios et al. Stanford university unstructured (su2): Open-source analysis and design technology for turbulent flows. *AIAA paper*, 243:13–17, 2014.

- [10] P. Moin and T. Bewley. Feedback control of turbulence. *Appl. Mech. Rev.*, 47(6):S3–S13, 1994.
- [11] P. Luchini, T. Bewley, and M. Quadrio. An optimal feedback controller for the reduction of turbulent energy in 3D plane-duct flow. In *APS Meeting, Chicago, IL (US), November 20-22*, 2005.
- [12] G.K. Karpouzas et al. Adjoint optimization for vehicle external aerodynamics. *International Journal of Automotive Engineering*, 7(1):1–7, 2016.
- [13] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245, 2010.
- [14] C. Othmer, E. de Villiers, and H.G. Weller. Implementation of a continuous adjoint for topology optimization of ducted flows. *AIAA Paper 2007-3947*, 2007.
- [15] Adjoint CFD technology at ENGYS. <http://engys.com/technology/adjoint-cfd>.
- [16] discreteAdjointOpenFOAM project at STCE, RWTH Aachen university. <https://www.stce.rwth-aachen.de/research/software/discreteadjointopenfoam>.
- [17] A. Quarteroni. *Modellistica numerica per problemi differenziali*, volume 97. Springer, 2016.
- [18] H. Strandenes. NACA2STL Matlab/Octave script. <https://github.com/pruthvi1991/Geometry/blob/master/NACA2STL.m>, 2012.
- [19] H. Strandenes. Meshgen Matlab/Octave script. <https://github.com/petebachant/NACAFoil-OpenFOAM/blob/master/scripts/meshgen.m>, 2012.
- [20] S. Sen, S. Mittal, and G. Biswas. Steady separated flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 620:89–119, 2009.
- [21] J. Brezillon and N.R Gauger. 2d and 3d aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8):715–727, 2004.