HAL
open science

Optimization of active control devices for separated turbulent flows

Jérémie Labroquère Labroquère

To cite this version:

Jérémie Labroquère Labroquère. Optimization of active control devices for separated turbulent flows. General Mathematics [math.GM]. Université Nice Sophia Antipolis, 2014. English. NNT : 2014NICE4096 . tel-01127187

HAL Id: tel-01127187

https://theses.hal.science/tel-01127187v1

Submitted on 7 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS
DOCTORAL SCHOOL SFA
SCIENCES FONDAMENTALES ET APPLIQUÉES

P H D T H E S I S

to obtain the title of

Doctor of Science

of the University of Nice - Sophia Antipolis
Speciality: Applied mathematics

Defended by

Jérémie Labroquère

Optimization of active control devices for separated turbulent flows

Thesis Advisors: Jean-Antoine Désidéri & Régis Duvigneau,
OPALE Project-Team

prepared at INRIA Sophia Antipolis, OPALE Projet-Team

defended on November 20th, 2014

Jury:

Reviewers:	Christian Tenaud	- DR -	CNRS UPR 3251 - LIMSI
	Michel Visonneau	- DR -	CNRS UMR 6598 - LMF - ECN
Advisors:	Jean-Antoine Désidéri	- DR -	INRIA - OPALE
	Régis Duvigneau	- CR -	INRIA - OPALE
Examinators:	Marianna Braza	- DR -	CNRS UMR 5502 - IMFT
	Rodolphe Le Riche	- CR -	CNRS UMR 6158 - LIMOS - EMSE

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

©2014 Jérémie Labroquère - jeremie.labroquere@gmail.com

Remerciements

J'aimerais remercier tout d'abord, les personnes sans qui cette thèse n'aurait pu voir le jour : Jean-Antoine Désidéri et Régis Duvigneau. Jean-Antoine m'a permis de prendre part dans l'équipe projet OPALE afin de mener ma recherche dans le domaine de l'optimisation et la simulation numérique. Il m'a été d'un grand apport concernant les aspects mathématiques fondamentaux, tout en partageant ses valeurs au regard de l'humanité. Régis Duvigneau a su être présent au jour le jour et a su m'apporter son recul sur les aspects numériques des mathématiques. Sans lui, je n'aurais jamais eu les outils, ni le courage pour mener à bien les études. Il m'a guidé tout au long de mon travail de thèses tout en corrigeant la dynamique et trajectoire qu'elle prenait. Un grand merci à lui, à sa patience exemplaire, à sa vision et la qualité de sa recherche qu'il m'a fait partager. Ce fut un grand moment de plaisir !

Je remercie également mes rapporteurs de thèse Christian Tenaud et Michel Visonneau pour avoir lu, compris et analysé ce manuscrit. Merci à eux, ainsi qu'à Marianna Braza et Rodolphe Le Riche d'avoir accepté de faire partie de mon Jury de thèse.

Je tiens à remercier l'équipe du LHEEA de l'École Central de Nantes, à savoir Ganbo Deng, Emmanuel Guilmineau, Patrick Queutey et Michel Visonneau pour m'avoir permis d'utiliser ISIS et conseillé sur les paramètres à utiliser pour les simulations.

Merci à mes collègues, anciennement doctorants de l'équipe projet OPALE, qui sont maintenant déjà en poste : Adrien Zerbinati pour les soirées à prendre l'apéro sur son balcon tout en buvant du bon vin et Fatima Zahra Oujebbour ma collègue de bureau qui a su me supporter et avec qui j'ai beaucoup échangé.

Merci à mes amis de longue date, pour ne pas dire d'enfance, avec qui j'ai des souvenirs merveilleux : Oryaëlle Chevrel, Romain Delpoux, Emeline Marsais, etc.

Merci à mes amis rencontrés à la danse, ou en Allemagne lors de ma dernière année de thèse qui ont réussi à me faire oublier plusieurs fois par semaines le travail nécessaire pour élaborer un tel manuscrit.

Enfin, un grand merci à ma mère, à mon père qui m'a toujours soutenu et épaulé dans les décisions que j'ai prises et à ma soeur pour m'avoir poussé dans les études dès le plus jeune âge.

Pour finir, merci à INRIA et à la direction, pour m'avoir ouvert ses portes. J'ai été impressionné par la qualité de cet institut d'exception, qui regorge de scientifiques de très haut niveau.

Acknowledgments

This thesis is partially supported by the 7th Framework Program of the European Union, project number 266326 ”MARS”.

Contents

I Introduction **1**

II Simulation of actuated flows **7**

1 Context and issues **9**

2 Fluid and actuation modeling **11**
2.1 Fluid mechanics equations . . . 11
2.1.1 Conservation laws . . . 12
2.1.2 Thermodynamic and perfect gas modeling . . . 13
2.1.3 Stress tensor modeling . . . 13
2.1.4 Heat flux model . . . 14
2.2 Filtered Navier-Stokes equations . . . 15
2.2.1 Turbulence phenomenon and computational burden . . . 15
2.2.2 Equations filtering . . . 16
2.3 Turbulence flow models . . . 19
2.3.1 Spalart-Allmaras . . . 20
2.3.2 k-ε Launder Sherma . . . 21
2.3.3 k-ω SST . . . 22
2.3.4 k-ω EASM . . . 24
2.4 Actuation modeling - Synthetic jet . . . 24
2.4.1 Cavity model . . . 25
2.4.2 Slot model . . . 26
2.4.3 Boundary condition model . . . 27

3 Numerical methods for simulation **29**
3.1 Num3sis - a platform for numerical simulation . . . 29
3.2 Mean turbulent flow . . . 31
3.3 Turbulent equations . . . 32
3.4 Space discretization . . . 33
3.4.1 Convective fluxes . . . 34
3.4.2 Viscous fluxes . . . 39
3.4.3 Source terms . . . 41
3.5 Time discretization . . . 41
3.6 Boundary conditions . . . 43
3.6.1 Weak approach for inlet, outlet and symmetry conditions . . . 43
3.6.2 Strong approach for wall and jet condition . . . 44
3.6.3 Strong iterative approach . . . 45
3.6.4 Viscous boundary contribution . . . 45

4 Validation for uncontrolled flows **49**

4.1 Turbulent subsonic flow over a flat plate 49
4.2 Turbulent subsonic flow around NACA0012 airfoil 51

5 Flow with actuation study 59
5.1 Comparison of the synthetic jet models 61
5.2 Impact of numerical parameters 80
5.3 Refinement study 81
5.4 Impact of turbulence closure 81
5.5 Synthesis 87

III Optimization algorithms 89

6 Context and issues 91

7 Methods 93
7.1 Meta-model based algorithms (Efficient Global Optimization) 93
7.1.1 Gaussian Process model 94
7.1.2 Correlation functions 97
7.1.3 Hyperparameters determination in Bayesian framework 98
7.1.4 Prediction with noisy observations 99
7.1.5 Construction of the Gaussian Process meta-model 101
7.1.6 Merit functions 101
7.2 Evolution strategy 103

8 Validation 107
8.1 1D analytical case 107
8.1.1 Deterministic case optimization 107
8.1.2 Optimization with noise 108
8.2 Noisy Branin function 110
8.2.1 Deterministic case optimization 112
8.2.2 Optimization with noise 112
8.3 Conclusion on optimization validation 114

IV Application to flow control 119

9 NACA0015 123
9.1 Test-case configuration 123
9.2 Synthetic jet optimization 130
9.2.1 Impact of observation variance 130
9.2.2 EI/PI comparison 132
9.2.3 Analysis of optimum controlled flows 136
9.3 Discussion 153

10 Backward Facing Step **155**
10.1 Incompressible turbulent Navier-Stokes equations . . . 155
10.2 Backward Facing Step . . . 156
10.2.1 Non controlled case . . . 158
10.2.2 Controlled case validation . . . 163
10.3 A naive optimization neglecting observation error . . . 170
10.4 Impact of observation variance . . . 171
10.5 Impact of turbulence closures . . . 173
10.5.1 Optimization results . . . 173
10.5.2 Cross-validation . . . 177
10.6 Discussion . . . 178

V Conclusion and prospects **183**

A Fluxes Jacobians **187**
A.1 Inviscid Jacobians . . . 187
A.1.1 General problem of finding the Jacobian for a numerical inviscid flux 187
A.1.2 Inviscid flux Jacobian . . . 189
A.2 Boundary Jacobians . . . 190
A.2.1 Decomposition of the Jacobian . . . 190
A.2.2 Symmetry condition . . . 191
A.2.3 Supersonic inlet . . . 192
A.2.4 Supersonic outlet . . . 192
A.2.5 Riemann Invariant subsonic inlet . . . 192
A.2.6 Riemann Invariant subsonic outlet . . . 195
A.3 Viscous Jacobians . . . 199
A.3.1 General problem of finding the Jacobian for a numerical viscous flux 199
A.3.2 Interior numerical viscous flux Jacobian computation . . . 200

Bibliography **207**

Part I

Introduction

Within the last decades, aviation industry literally exploded and became highly competitive. Since, new problems such as human activity environment threatening and global warming rose up. In Europe, this issue has been defined as one of the highest priorities in the aviation industry for the next 40 years. Indeed, relative to the capabilities of a typical new aircraft in 2000, the Flightpath 2050 Europe's Vision for Aviation targets a 75% reduction in CO_2 emissions, 90% in NO_x emissions and 65% of the perceived noise emission of flying aircrafts [Commission 2011]. To achieve such reductions, increasing the efficiency of current technologies will certainly not be sufficient.

New technology such as active flow control has been pointed out by industries, such as Airbus, as a key enabling technology to match the requirements of gas emissions by 2050. Flow control consists in altering the state of a natural flow to reach a new state characterized by interesting properties. In other words, actuators are introduced in a flow to obtain some desirable characteristics. This is not a recent idea as, at the beginning of the last century, Ludwig Prandtl applied flow control to explain his boundary layer theory [Prandtl 1904] by completely suppressing boundary layer separation past a circular cylinder using constant suction. Fig. 1 illustrates an active flow control strategy to attach the flow around an airfoil, which naturally generates vortex shedding [Glezer 2004, Amitay 2001].

Flow control can be passive or active. Passive flow control is achieved using actuators that do not involve any external power to be driven. These actuators are typically fixed mechanical facilities acting on the boundary layer, such as surface riblets, vortex generators, etc. They are easy to implement but their efficiency is limited and they suffer from a lack of flexibility.

Active flow control, on the other hand, needs to supply additional energy. Steady momentum injection, fluid blowing or suction [Prandtl 1904, Hoarau 2006] are part of this category. Unfortunately, steady fluid blowing has a low efficiency and requires air supply systems, keeping it away from practical industrial use. Alternately, unsteady active actuators exploit natural flow instability phenomena and have been a growing research area for the last decades, since their ability to improve aerodynamic performance [Gad-el Hak 1998] was demonstrated. Effective control can be achieved using periodic actuators to interact with the large-scale coherent structures. These periodic flow excitations may be driven using synthetic jets, plasma actuators, dielectric barrier discharge, etc. Several investigations have been carried out, experimentally as well as numerically, to apply this technology for various purposes, such as separation delay [Seifert 1996], mixing enhancement [Ritchie 2000], flow vectorization [Smith 1997], etc.

A major difficulty in active flow control is the choice of control parameters such as amplitude, frequency, as illustrated in Fig. 1. Furthermore, doing everything with experimental studies on such large scale is inconceivable. On a real aircraft, hundreds of actuators would be necessary and doing the choice by hand is not possible.

Lately, numerical capabilities and computational fluid dynamics algorithms have reached a maturity level allowing numerical experiments to replace regular wind tunnel campaigns

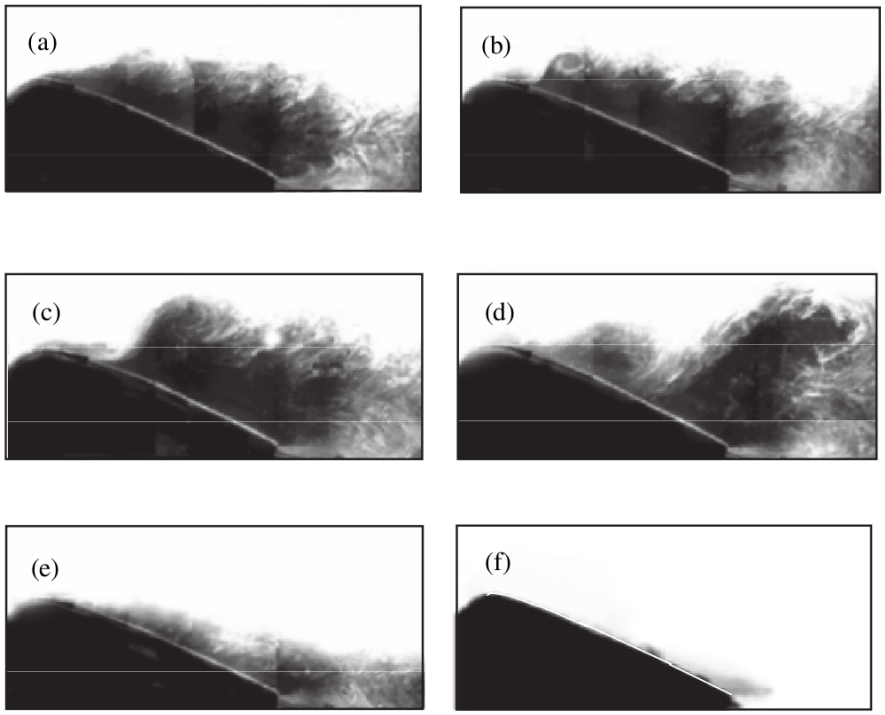


Figure 1: Phase-averaged smoke-visualization images of the flow above the surface of a stalled airfoil: (a) baseline characterized by a large recirculating area (in grey), (b) to (f) periodic actuation using suction/blowing at the leading edge for different frequencies. Actuation generates various flow patterns, the flow being completely re-attached for case (f). [Glezer 2004]

and postponing models validation at a later stage in the design process. As the aim is to exploit natural instability phenomena to manipulate flow characteristics, active flow control is very sensitive to actuator parameters and finding the best effective parameters actuators is thus a requirement. While numerous studies have been focused on parametric studies [Ekaterinaris 2003], advanced optimization algorithms are gaining an interest as they can assist in determining such parameters. By coupling simulation of actuated flows with optimization algorithms, one could hope to determine efficient control parameters for a reasonable cost.

Unfortunately, simulation is still limited. The first difficulty is related to turbulence modeling. Three levels of turbulence modeling are used in practice. The first one is the DNS, for Direct Numerical Simulation [Spalart 1986]. It makes no assumption about turbulence physics and solve the smallest possible eddies in the flow. This level of modeling is computationally intractable for the considered industrial flows due to grid requirements. The second level of modeling is the LES/DES, for Large Eddy Simulation [Lesieur 2005, Sagaut 2002] or Detached Eddy Simulation [Spalart 2009]. They model small scales of turbulent eddies and make no assumption about the large scales. These two models are more affordable in terms of CPU cost, but their requirements are

still in the order of a few weeks for only one simulation. Furthermore, their applicability in industry is still problematic because of requirements in terms of grid and computational cost. The last level of modeling is the RANS, for Reynolds Averaged Navier-Stokes equations. This modeling technique consists in applying a filtering operator to the Navier-Stokes equations to solve the mean flow only. This model is the most affordable one and is commonly used in industry. The main drawback is that most of the time, the physics is not well reproduced when the simulations are run too far away from calibration cases. Despite of these drawbacks, many simulations of unsteady controlled flows using RANS models can be found in the literature [Donovan 1998, Ekaterinaris 2003].

Regarding optimization, the main difficulties in the context of turbulent flow control is the CPU time and possible local optima. Gradient based optimization methods such as the simple steepest descent method, can reduce the number of necessary evaluations for a complete optimization. That leads to a low CPU time but these methods are known to be trapped in local optima. In addition, computing objective function gradients with respect to control parameters, because of solver evaluation cost, excludes the use of finite differencing. Adjoint approaches are typically used to efficiently compute such quantities with steady flows, but they become really cumbersome and difficult to apply in case of unsteady flows [Belme 2011]. Meta-heuristic methods (Evolution Strategy [Michalewicz 1996], Genetic Algorithm [Goldberg 1989], Particle Swarm Optimization [Eberhart 1995]) can reach global optima but are too costly as they require hundreds to thousands of function evaluations.

Therefore, the objective of this thesis is to explore the feasibility of an alternative global optimization technique based on unsteady simulations to determine actuation parameters in the context of turbulent flows. More precisely, we will focus on the Efficient Global Optimization (EGO) [Jones 1998] approach, which is based on an adaptive statistical modeling of the cost functional, allowing to account for observation errors. Moreover, a particular attention will be paid to the impact of the physical modeling level in the optimization process. One intends to assess the capability of RANS models to determine efficient control parameters in this context.

This manuscript is structured as follows. Chapter 1 presents the computational fluid dynamics methods with an optimization perspective. Chapter 2 provides the description of the optimization procedure and theory with the perspective of optimizing functions evaluated using highly costly simulations. Chapter 3 presents the application of the methods on industrial relevant problems of active flow control (proposed in the framework of MARS project).

Part II

Simulation of actuated flows

Chapter 1

Context and issues

Simulation of actuator-induced flows is still a challenge nowadays. The encountered difficulties result from the large ranges of space and time scales involved in actuated flow simulations. Indeed, different physical scales are involved near the actuator and within the separated regions. Turbulence phenomenon further complicates the problem as it introduces strong requirements in terms of computational power and modeling, leading to uncertainties in the simulations. Turbulent flows are usually solved using Reynolds Averaged Navier-Stokes (RANS) equations, Large-Eddy Simulation (LES) or Direct Numerical Simulation (DNS). These three approaches have an increasing precision and computational cost. The DNS method denotes solutions of the Navier-Stokes equations with all space-time scales resolved. It is the most accurate method but is simply not applicable at high Reynolds numbers and on complex problems. On the other hand, LES involves spatially filtered variables depending on the grid resolution and usually outclass RANS models for such problems. However, LES methods are still unable to predict actuated flows in a computational time that would be reasonable in an industrial context. RANS methods use time filtering allowing computation at a lower expense, but the resulting models are calibrated on typical flow problems, which make them usually deficient when the flow conditions are outside the range of calibration. The reason why the cost of simulation is a big concern with an optimization perspective is that, in average, solving an optimization problem with a gradient free method takes at least dozens of unsteady computations. This excludes the use of complex turbulence models such as LES. However, in a recent study [Garnier 2012], four RANS and one LES models have been compared in the case of a separated flow including a synthetic jet actuator, for a set of thirteen frequencies. It has been found that RANS models are not in a position to predict the correct characteristics of the separation, but some of them are able to identify the best actuation frequency, among those tested, to reduce the separation length. The decrease of the recirculation length is under-estimated by RANS models, but the ranking of the actuation parameters has been correctly established. This encouraging result indicates that RANS models could possibly be employed in a design phase to select control parameters. An objective of this thesis is to establish if RANS models can be used in an optimization framework, which is more ambitious.

In this thesis, the main interest is focused on synthetic jet actuators only. Synthetic jets are devices which periodically suck and blow surrounding fluid from a slot usually triggered by a diaphragm placed at the bottom of a cavity. This periodical fluid movement locally perturbs the flow and allows to manipulate some characteristics of the flow. They

have been extensively studied, and they represent one of the most promising technologies in the near future for flow control applications. Apart from the problem of turbulence closures for detached flows, it has been demonstrated, in particular during the CFDVAL Workshop [Rumsey 2006], that the prediction of actuator-induced flows is still tedious, because of the actuator model itself. In [Yamaleev 2005], N. K. Yamaleev makes a review of multiple possibilities for numerical synthetic jets modeling and it is shown that a common consensus is not yet established. Moreover, cost to simulate the actuator can be non negligible because of the mesh refinement and the internal description of the actuator. A wide range of simplified models have been developed and studied, such as:

- imposing simplified boundary conditions at the orifice exit,
- partially modeling the problem including the synthetic jet slot, and applying simple boundary condition,
- modeling the entire problem including the cavity and the synthetic jet slot, and applying simple boundary condition at the cavity bottom,
- modeling the orifice actuator using unsteady pipe-flow theory,
- using ordinary differential equations describing a resonant fluid actuator,
- applying a reduced order model using the lumped element model simulating a piezoelectric-driven synthetic jet actuator,
- describing the entire actuator geometry and simulating the diaphragm as a moving boundary,
- using localized body forces along desired points in the computational mesh.

All of these models of various fidelity have advantages and drawbacks. For example, the full description of the fluid actuator employing a moving boundary is too costly in terms of computational resources. Especially, in the perspective of considering a large number of actuators. At a reduced cost, the reduced order model and the ordinary differential equations approaches can give satisfactory results but might lack some features to truly represent the actuator [Rumsey 2006]. In this thesis, we focus on the three first models, based on applying simplified boundary conditions with an increasing complexity in terms of geometrical representation. A second objective of this thesis is thus to quantify the impact of the actuator model, in the perspective of design optimization.

In the following, the mathematical model and the underlying hypothesis for the simulation of compressible turbulent flows are first described. Then different turbulence models are exposed, and an overview of actuation modeling is given.

Chapter 2

Fluid and actuation modeling

Contents

2.1 Fluid mechanics equations **11**

2.1.1 Conservation laws 12

2.1.2 Thermodynamic and perfect gas modeling 13

2.1.3 Stress tensor modeling 13

2.1.4 Heat flux model 14

2.2 Filtered Navier-Stokes equations **15**

2.2.1 Turbulence phenomenon and computational burden 15

2.2.2 Equations filtering 16

2.3 Turbulence flow models **19**

2.3.1 Spalart-Allmaras 20

2.3.2 k-ε Launder Sherma 21

2.3.3 k-ω SST 22

2.3.4 k-ω EASM 24

2.4 Actuation modeling - Synthetic jet **24**

2.4.1 Cavity model 25

2.4.2 Slot model 26

2.4.3 Boundary condition model 27

2.1 Fluid mechanics equations

Fluid mechanics equations are based on the continuum hypothesis considering a fluid as a continuous medium instead of composed of discrete molecules. This assumption is true under the condition that the Knudsen number, defined as the ratio of molecule mean free path length to a representative length scale (i.e: the discretization length scale), is lower than 0.01. Typical aerodynamic flows considered here satisfy this condition.

2.1.1 Conservation laws

The Navier-Stokes equations are the macroscopic view of fluid mechanics equations based on this continuum hypothesis. These partial differential equations are established considering *conservation of mass*, *momentum* and *energy* principles. The partial differential equations describe the temporal evolution of definite quantities due to the convection, the diffusion, the surface sources and the volume sources. A fluid at the time t and position $\boldsymbol{x}$, in the domain Ω, can be described by its density $\rho(\boldsymbol{x}, t)$, velocity $\boldsymbol{V}(\boldsymbol{x}, t)$, pressure $p(\boldsymbol{x}, t)$ and temperature $T(\boldsymbol{x}, t)$ fields. By assuming that the rate of change of mass of the fluid is equal to the mass fluxes, the conservation of mass equation is derived. The conservation of mass is written as:

$$\partial_t \rho + \boldsymbol{\nabla}(\underbrace{\rho \boldsymbol{V}}_{\boldsymbol{F_c}}) = 0. \tag{2.1}$$

The second conservation equation, as a result of Newton's law, stipulates that the momentum rate of is the consequence of the net force applied to it. Here, the external forces are split into two main groups: the volume forces, denoted by $\boldsymbol{f_e}$, and the surface forces defined by the stress tensor $\boldsymbol{\sigma}$. The equation of momentum conservation is written as:

$$\partial_t(\rho \boldsymbol{V}) + \boldsymbol{\nabla}(\underbrace{\rho \boldsymbol{V} \otimes \boldsymbol{V}}_{\boldsymbol{F_c}}) = \boldsymbol{\nabla}(\underbrace{\boldsymbol{\sigma}}_{\boldsymbol{Q_s}}) + \underbrace{\rho \boldsymbol{f_e}}_{\boldsymbol{Q_v}}, \tag{2.2}$$

with $\boldsymbol{\sigma}$ the stress tensor, $\boldsymbol{V} = (u, v, w)^{\mathrm{T}}$ the velocity, $\boldsymbol{V} \otimes \boldsymbol{V}$ the tensorial product of $\boldsymbol{V}$:

$$\boldsymbol{V} \otimes \boldsymbol{V} = \begin{pmatrix} u^2 & uv & uw \\ vu & v^2 & vw \\ wu & wv & w^2 \end{pmatrix}. \tag{2.3}$$

Finally, the last Navier-Stokes equation is the conservation of energy. The rate of variation of energy depends on the work of external and internal forces and heat flux. The conservation of energy is written as:

$$\partial_t(\rho E) + \boldsymbol{\nabla}(\underbrace{(\rho E)\boldsymbol{V}}_{\boldsymbol{F_c}}) = \boldsymbol{\nabla}(\underbrace{\boldsymbol{q}}_{\boldsymbol{F_d} = -K\boldsymbol{\nabla}(\boldsymbol{e})}) + \boldsymbol{\nabla}(\underbrace{\boldsymbol{\sigma} \cdot \boldsymbol{V}}_{\boldsymbol{Q_s}}) + \underbrace{\rho(\boldsymbol{f_e} \cdot \boldsymbol{V}) + q_H}_{\boldsymbol{Q_v}} \tag{2.4}$$

with $\boldsymbol{q}$ the heat flux, q_H the heat volume source, the total energy by unity of mass $E = e + \frac{1}{2}\boldsymbol{V} \cdot \boldsymbol{V}$ [1] and e the internal energy. We refer the reader to specialized references [Hirsch 2007] for the establishment of such equations.

To close the Navier-Stokes equations, one needs to model the stress tensor $\boldsymbol{\sigma}$, the heat flux $\boldsymbol{q}$ and to link them with the state variables. This is the purpose of the following subsections.

[1] Note that for some authors, the total energy by unity of volume is used

2.1.2 Thermodynamic and perfect gas modeling

The perfect gas model, established upon observations, expresses the relationship between the pressure, the volume and the temperature. It is valid for a wide range of gazes and is written as:

$$p = \rho r T$$

with r a constant depending on the considered gas. Typically, for the air in normal conditions, $r = 287 Jkg^{-1}K^{-1}$.

To complete the description of a gas, one needs to specify the capacity of a gas to absorb or release energy. This property is given by the specific heat capacity coefficient at constant volume $\mathrm{c_v}$ and the heat coefficient at constant pressure $\mathrm{c_p}$ in $Jkg^{-1}K^{-1}$. They describe the variation of energy of $1kg$ of fluid when the temperature is raised or decreased by $1K$ under a constant volume or pressure. For a perfect gas, the specific heat capacity at constant pressure is:

$$\mathrm{c_p} = \frac{\gamma r}{\gamma - 1}.$$

The Meyers' relations complete the description of the perfect gas by linking the previous two coefficients, with the constant:

$$\gamma = \frac{\mathrm{c_p}}{\mathrm{c_v}}.$$

From this, the internal energy e and sound speed c are derived using:

$$e = \mathrm{c_v} T = \frac{1}{\gamma - 1}\frac{p}{\rho},$$

$$c = \sqrt{\gamma r T} = \sqrt{\frac{\gamma p}{\rho}}.$$

2.1.3 Stress tensor modeling

For a solid, the stress tensor is related to the material deformation, or strain tensor. A strain-stress $\boldsymbol{\sigma}(\mathbf{E}, ...)$ relation can be observed from experiments. For a fluid, a strain rate-stress relation $\boldsymbol{\sigma}(\mathbf{D}, ...)$ is observed, with $\mathbf{D}$ the strain rate tensor defined as:

$$\mathbf{D} = \frac{1}{2}\left(\boldsymbol{\nabla V} + (\boldsymbol{\nabla V})^T\right). \tag{2.5}$$

The stress tensor $\boldsymbol{\sigma}$ can be decomposed in a *spherical* and a *deviator* part, respectively representing dilatation and distortion. Considering a fluid particle, the deviator tensor, which models a stress independent of the rotation, comes from the hydrostatic pressure and the deviator tensor from viscous forces. Thus, the stress tensor is written as:

$$\boldsymbol{\sigma} = \underbrace{\boldsymbol{\tau}}_{\text{Deviator tensor}} - \underbrace{p\boldsymbol{I_n}}_{\text{Spheric tensor}} \tag{2.6}$$

The pressure in the spheric tensor is expressed using a state law equation while the deviator tensor with material law properties. The component τ_{ij} of the shear stress tensor $\boldsymbol{\tau}$ can be seen as:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} \tag{2.7}$$

2.1.3.1 Newtonian fluid model

The Newtonian fluid model assumes a shear stress tensor $\boldsymbol{\tau}$ proportional to the strain rate tensor $\mathbf{D}$, or velocity gradient. The involved proportional coefficient is the *viscosity* μ. Under homogeneous and isotropic medium assumption, the shear stress tensor only depends on the local thermodynamic state:

$$\boldsymbol{\tau} = 2\mu\mathbf{D} - \frac{1}{3}(\mathrm{tr}(2\mu\mathbf{D}))\boldsymbol{I_n} \tag{2.8}$$

$$\tau_{xx} = \frac{2}{3}\mu(2\partial_x u - \partial_y v - \partial_z w), \quad \tau_{yy} = \frac{2}{3}\mu(2\partial_y v - \partial_z w - \partial_x u), \quad \tau_{zz} = \frac{2}{3}\mu(2\partial_z w - \partial_x u - \partial_y v),$$

$$\tau_{xy} = \mu(\partial_y u - \partial_x v), \quad \tau_{yz} = \mu(\partial_z v - \partial_y w), \quad \tau_{zx} = \mu(\partial_x w - \partial_z u).$$

2.1.3.2 Viscosity model

For simple cases, the viscosity is usually taken as a constant. For more complex cases, where the range of temperature and pressure is large, state variables such as the temperature can be involved. As long as the temperature is kept reasonably low, a typical viscosity model is defined by the *Sutherland law*:

$$\mu(T) = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^{3/2}\frac{T_{ref}+T_s}{T+T_s},$$

with the reference viscosity $\mu_{ref} = 1.716 \cdot 10^{-5}\, N.s.m^{-2}$, the reference temperature $T_{ref} = 273.15\, K$ and the Sutherland temperature $T_s = 110.4\, K$.

2.1.4 Heat flux model

The remaining unknown in the equations is the heat flux $\boldsymbol{q}$. The heat flux is modeled by a function of the temperature gradient $\boldsymbol{\nabla} T$ and material thermal conductivity κ in $WK^{-1}m^{-1}$:

$$\boldsymbol{q}(\boldsymbol{\nabla} T, \kappa) = -\kappa \boldsymbol{\nabla} T$$

i.e:

$$\boldsymbol{q} = (q_x, q_y, q_z)^T = -\kappa\,(\partial_x T, \partial_y T, \partial_z T)^T$$

The conductivity κ is a function of the material properties. For our purpose, a dimensionless number Pr, called the *Prandtl* number, is introduced and expresses the ratio of momentum diffusivity (μ/ρ) to the thermal diffusivity ($\kappa/(\rho \mathrm{c_p})$):

$$\mathrm{Pr} = \frac{\mathrm{c_p}\mu}{\kappa},$$

with $\mathrm{c_p}$ the specific heat at constant pressure, μ the fluid viscosity and κ the thermal conductivity. From this equation, the thermal conductivity is extracted and used to define the heat flux:

$$\kappa = \frac{\mathrm{c_p}\mu}{\mathrm{Pr}}.$$

The Prandtl number is usually taken as a constant $\mathrm{Pr} = 0.72$ but the following formula can also be used [Toro 2009]:

$$\mathrm{Pr}(\gamma) = \frac{4\gamma}{9\gamma - 5}$$

2.2 Filtered Navier-Stokes equations

Flow behaviors have been studied by Osborne Reynolds in details in the late 18th century. He extracted a very important non-dimensional number which characterize the complexity of a flow. This number, called the *Reynolds number*, is the ratio of the inertial forces over the viscous forces, or the ratio of viscous to convective time scales:

$$Re = \underbrace{\frac{\rho_{ref}\|\boldsymbol{V}_\infty\|^2 L^2}{\mu_{ref}\|\boldsymbol{V}_\infty\| L_{ref}}}_{\text{forces ratio}} = \underbrace{\frac{\rho_{ref} L_{ref}^2/\mu_{ref}}{L/\|\boldsymbol{V}_\infty\|}}_{\text{time scales ratio}} = \frac{\rho_{ref}\|\boldsymbol{V}_\infty\| L_{ref}}{\mu_{ref}},$$

with a reference density ρ_{ref}, a conventional characteristic length L_{ref}, the mean infinite velocity $\boldsymbol{V}_\infty$ and a reference viscosity μ_{ref}. When the Reynolds number is small, viscous forces are dominating the flow and the resulting flow is *laminar*. When the Reynolds number is large, inertial forces are dominating the flow and the flow becomes *turbulent*. In between, at moderate Reynolds number, the flow has laminar-turbulence intermittency behavior and is much more difficult to characterize.

With some simplifications and simple configurations, analytical solutions of the Navier-Stokes equations can be derived. Such solutions for incompressible laminar flows on Poiseuille (flow in a tube) or Blasius (flow over a plate) configuration are well known. Unfortunately, no general analytical solution of the Navier-Stokes solutions exists, especially in the turbulent regime. As a consequence, numerical computation is the only way to approach these solutions.

2.2.1 Turbulence phenomenon and computational burden

In most of engineering cases, even with the best computer/cluster, the *Direct Numerical Simulation (DNS)* where the smallest eddies are solved, is out of reach because the

grid has to be extremely fine to capture such eddies. If we assume the exponential growth of supercomputing power or Moore's Law holds, DNS computation in external aerodynamics will only be viable in 2080 [Spalart 2012]. Some turbulence modeling techniques, as pure *Large Eddy Simulation* (LES) where the influence of eddies smaller than the grid size is modeled, can reduce this date down to 2045. Fortunately, some other turbulence modeling techniques that are computationally tractable exist, but their domain of validity and underlying hypothesis can be a major concern in the optimization framework. These models are the ones we focus on in this thesis as they are the only ones that are compatible with the required computational time for the optimization.

2.2.2 Equations filtering

To reduce the computational cost of solving turbulent flows, one idea is to solve only a part of the flow and to model what remains. To do so, the Navier-Stokes equations are filtered using an operator $< . >$. *Statistical methods* aim to solve the mean flow and to model fluctuations coming from the turbulence. The main properties of this filter are:

- constants conservation: $< c >= c$
- linearity: $< f + g >=< f > + < g >$
- commutation with derivative (spatial and time): $\partial_s < f >=< \partial_s f >$, $s = \{\boldsymbol{x}, t\}$

with f and g two random variables and c a constant.

With such a filter, any variable x can be decomposed in a resolved and an unresolved part. In example, for the mean filter, $x = \overline{x} + x'$ with $\overline{x}$ the mean or resolved part and x' the fluctuations or unresolved part of x.

An examples of filter commonly used in turbulence modeling is the statistical mean filter:

$$< f(\boldsymbol{x}, t) >= \lim_{T \to \infty} \frac{1}{T} \int_{t0}^{t0+T} f(\boldsymbol{x}, t) dt$$

By applying such a filter to the Navier-Stokes equations [Mohammadi 1994], the conservation of mass becomes:

$$\begin{aligned} < \partial_t \rho + \boldsymbol{\nabla} \rho \boldsymbol{V} > = \\ \partial_t < \rho > + \boldsymbol{\nabla} < \rho \boldsymbol{V} > = 0. \end{aligned} \tag{2.9}$$

The conservation of momentum:

$$\begin{aligned} < \partial_t (\rho \boldsymbol{V}) + \boldsymbol{\nabla}(\rho \boldsymbol{V} \otimes \boldsymbol{V}) - \boldsymbol{\nabla}(\boldsymbol{\sigma}) - \rho \boldsymbol{f_e} > = \\ \partial_t (< \rho \boldsymbol{V} >) + \boldsymbol{\nabla}(< \rho \boldsymbol{V} \otimes \boldsymbol{V} > - < \boldsymbol{\sigma} >) - < \rho \boldsymbol{f_e} > = 0. \end{aligned} \tag{2.10}$$

And the conservation of energy:

$$\begin{aligned} < \partial_t (\rho E) + \boldsymbol{\nabla}((\rho E)\boldsymbol{V}) - \boldsymbol{\nabla}(\boldsymbol{q}) - \boldsymbol{\nabla}(\boldsymbol{\sigma} \cdot \boldsymbol{V}) - \rho(\boldsymbol{f_e} \cdot \boldsymbol{V}) - q_H > = \\ \partial_t < \rho E > + \boldsymbol{\nabla}(< (\rho E)\boldsymbol{V} > - < \boldsymbol{q} > - < \boldsymbol{\sigma} \cdot \boldsymbol{V} >) - < \rho(\boldsymbol{f_e} \cdot \boldsymbol{V}) > - < q_H > = 0. \end{aligned} \tag{2.11}$$

To simplify the set of variables, the Favre density-weighted filtering operation is introduced:

$$\widetilde{A} =< \rho A > / < \rho >,$$

with the associated non resolved part $A'' = \widetilde{A} - A$. The filtered Navier-Stokes equations then read:

$$\begin{cases} \partial_t(< \rho >) + \boldsymbol{\nabla}(< \rho > \widetilde{\boldsymbol{V}}) = 0 \\ \partial_t(< \rho > \widetilde{\boldsymbol{V}}) + \boldsymbol{\nabla}(< \rho \boldsymbol{V} \otimes \boldsymbol{V} > - < \boldsymbol{\sigma} >) - < \rho > \widetilde{\boldsymbol{f_e}} = 0 \\ \partial_t(< \rho > \widetilde{E}) + \boldsymbol{\nabla}(< \rho E \boldsymbol{V} > - < \boldsymbol{q} > - < \boldsymbol{\sigma} \cdot \boldsymbol{V} >) - < \rho > (\widetilde{\boldsymbol{f_e} \cdot \boldsymbol{V}}) - < q_H >= 0 \end{cases}$$

It remains to develop the different parts of the equations using the filter properties by focusing respectively on the terms $< \rho \boldsymbol{V} \otimes \boldsymbol{V} >$, $< \rho E \boldsymbol{V} >$ and $< \boldsymbol{\sigma} \cdot \boldsymbol{V} >$. It can be shown using the filters properties that $< \rho \boldsymbol{V} \otimes \boldsymbol{V} >$ is decomposed as:

$$< \rho \boldsymbol{V} \otimes \boldsymbol{V} >=< \rho > \widetilde{\boldsymbol{V}} \otimes \widetilde{\boldsymbol{V}} + < \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' > .$$

With this decomposition, the *Reynolds stress tensor* $\boldsymbol{R} = - < \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >$ is naturally derived.

In the same manner, using the filtering properties, $< \rho E \boldsymbol{V} >$ is decomposed as:

$$< \rho E \boldsymbol{V} >=< \rho > \widetilde{E} \widetilde{\boldsymbol{V}} + < \rho E'' \boldsymbol{V}'' > .$$

The turbulent transport of E'' represented by the term $< \rho E'' \boldsymbol{V}'' >$ needs to be further developed by expressing the term E'' with $\widetilde{E}$ and E. The expression of $\widetilde{E}$ is found by using the total energy definition:

$$E = e + \frac{1}{2} \boldsymbol{V} \cdot \boldsymbol{V}, \tag{2.12}$$

and by averaging it:

$$\begin{aligned} < \rho E > &=< \rho e > + \frac{1}{2} < \rho \boldsymbol{V} \cdot \boldsymbol{V} > \\ =< \rho > \widetilde{E} &=< \rho > \widetilde{e} + \frac{1}{2} < \rho \widetilde{\boldsymbol{V}} \cdot \widetilde{\boldsymbol{V}} > + \frac{1}{2} < \rho \boldsymbol{V''} \cdot \boldsymbol{V''} > \\ \Leftrightarrow \widetilde{E} &= \widetilde{e} + \frac{1}{2} \frac{< \rho \widetilde{\boldsymbol{V}} \cdot \widetilde{\boldsymbol{V}} >}{< \rho >} + \frac{1}{2} \frac{< \rho \boldsymbol{V''} \cdot \boldsymbol{V''} >}{< \rho >} \end{aligned} \tag{2.13}$$

Thus

$$\widetilde{E} = \widetilde{e} + \frac{1}{2} \widetilde{\boldsymbol{V}} \cdot \widetilde{\boldsymbol{V}} + \underbrace{\frac{1}{2} \widetilde{\boldsymbol{V''} \cdot \boldsymbol{V''}}}_{\text{Turbulent kinetic energy}} \tag{2.14}$$

which is the definition of the Favre filtered total energy. From this averaging, the turbulent kinetic energy $\frac{1}{2} \widetilde{\boldsymbol{V''} \cdot \boldsymbol{V''}}$ is naturally showing up.

The term $< \rho E'' \boldsymbol{V}'' >$ can finally be developed by expressing E'':

$$\begin{aligned} E'' &= E - \widetilde{E} \\ &= (e - \widetilde{e}) + \frac{1}{2}\boldsymbol{V}\cdot\boldsymbol{V} - \frac{1}{2}\widetilde{\boldsymbol{V}}\cdot\widetilde{\boldsymbol{V}} - \frac{1}{2}\widetilde{\boldsymbol{V}''\cdot\boldsymbol{V}''} \\ &= e'' + \frac{1}{2}(\widetilde{\boldsymbol{V}} + \boldsymbol{V}'')\cdot(\widetilde{\boldsymbol{V}} + \boldsymbol{V}'') - \frac{1}{2}\widetilde{\boldsymbol{V}}\cdot\widetilde{\boldsymbol{V}} - \frac{1}{2}\widetilde{\boldsymbol{V}''\cdot\boldsymbol{V}''} \\ &= e'' + \widetilde{\boldsymbol{V}}\cdot\boldsymbol{V}'' + \frac{1}{2}(\boldsymbol{V}''\cdot\boldsymbol{V}'' - \widetilde{\boldsymbol{V}''\cdot\boldsymbol{V}''}) \end{aligned} \tag{2.15}$$

And with some algebra manipulation:

$$< \rho E'' \boldsymbol{V}'' > = < \rho e'' \boldsymbol{V}'' > + \widetilde{\boldsymbol{V}} \underbrace{< \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >}_{-\boldsymbol{R}} + \underbrace{\frac{1}{2} < \rho \boldsymbol{V}''(\boldsymbol{V}''\cdot\boldsymbol{V}'') >}_{\text{Neglected}}$$

The term $< \rho \boldsymbol{V}''(\boldsymbol{V}''\cdot\boldsymbol{V}'') >$ is considered to be small compared to the other terms and is thus neglected in practice.

Finally, the averaged conservation of mass is

$$\partial_t < \rho > + \nabla(< \rho > \widetilde{\boldsymbol{V}}) = 0, \tag{2.16}$$

the averaged conservation of momentum

$$\partial_t(< \rho > \widetilde{\boldsymbol{V}}) + \nabla(< \rho > \widetilde{\boldsymbol{V}} \otimes \widetilde{\boldsymbol{V}} + \underbrace{< \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >}_{\text{Reynolds stress tensor}}) = \nabla < \boldsymbol{\sigma} > + < \rho > \widetilde{\boldsymbol{f_e}}, \tag{2.17}$$

and the averaged conservation of energy

$$\begin{aligned} &\partial_t(< \rho > \widetilde{E}) + \nabla(< \rho > \widetilde{E}\widetilde{\boldsymbol{V}} + \underbrace{< \rho e'' \boldsymbol{V}'' >}_{-K_t \boldsymbol{\nabla}(\widetilde{e})} + \widetilde{\boldsymbol{V}} \underbrace{< \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >}_{\text{Reynolds stress tensor}} \\ &- \underbrace{< \boldsymbol{q} >}_{\boldsymbol{Fd} = -\kappa \boldsymbol{\nabla}(\widetilde{e})} - < \boldsymbol{\sigma}\cdot\boldsymbol{V} >) = < \rho > (\widetilde{\boldsymbol{V}\cdot\boldsymbol{f_e}}) + < q_H > . \end{aligned} \tag{2.18}$$

These filtered equations are formally identical to the non filtered Navier-Stokes equations with two terms in addition. The term $< \rho e'' \boldsymbol{V}'' >$ is modeled as a turbulent heat flux:

$$< \rho e'' \boldsymbol{V}'' > = -\kappa_t \boldsymbol{\nabla}(\widetilde{e}).$$

with κ_t the thermal turbulent conductivity. The Reynolds stress tensor $< \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >$ needs to be modeled to close the equations.

2.2.2.1 RANS and URANS approach

The Reynolds Averaged Navier-Stokes (RANS) equations and Unsteady Reynolds Averaged Navier-Stokes (URANS) equations are derived from this formal Navier-Stokes

equations filtering. RANS equations consider a time average filter applied to the Navier-Stokes equations:

$$< f(\boldsymbol{x}, t) >= \lim_{T \to \infty} \frac{1}{T} \int_{t0}^{t0+T} f(\boldsymbol{x}, t) dt.$$

With this filter, the statistical averaging is similar to time averaging under ergodicity hypothesis.

Unfortunately, this last hypothesis doesn't hold with non stationary flows, which are the ones in our concern. Thus, instead of using a time average filter, phase average filtering is preferred. The URANS equations are identical to the RANS equations but explicitly including a time dependance when solving the equations. This filter is defined by the mean of a convolution[Gatski 2007]:

$$< f(\boldsymbol{x}, t) >= \int \mathcal{G}(\boldsymbol{x} - \boldsymbol{r}, t - \tau) f(x, t) d\boldsymbol{r} d\tau,$$

where

$$\mathcal{G}(x, t) = \delta(x) \left[\lim_{N \to \infty} \frac{1}{N+1} \sum_{n=0}^{N} \delta(t + nT) \right].$$

with δ the Dirac delta function and T is the period of the cycle considered. This filter assumes that the averaged quantity is statistically periodic in time. This is questionable but can be reasonably assumed if the studied flow is excited with a periodic source.

To close the filtered Navier-Stokes equations, one needs to model the reynolds stress tensor and link it to known variables. The following describes some turbulence models used to close the equations.

2.3 Turbulence flow models

Turbulence modeling consists in defining the Reynolds stress tensor to close the resulting equations from the filtering process. Three main families exist within the RANS framework to model this term: the linear eddy viscosity models, the non linear eddy viscosity models and the Reynolds Stress Models. The linear and non linear eddy viscosity models use a relationship involving the mean strain rate, the vorticity fields and some other turbulence variables to the Reynolds Stress Tensor:

$$\boldsymbol{R} = - < \rho \boldsymbol{V}'' \otimes \boldsymbol{V}'' >= f\left(\mathbf{D}_{ij}, \boldsymbol{w}_{ij}, \mu_t, k, \dots\right)$$

with the strain rate tensor $\mathbf{D}_{ij}$, the vorticity tensor $\boldsymbol{w}_{ij}$, the turbulent viscosity μ_t and the turbulent energy k. On the other side, Reynolds Stress Models do not make assumption on the structure of the Reynolds stress tensor but uses transport equations to model its evolution.

Similar to the Newtonian fluid model defining the shear stress tensor, the linear eddy viscosity models, based on the so-called *Boussinesq eddy viscosity assumption*, define the

Reynolds stress tensor as:

$$\boldsymbol{R} = \left(2\mu_t \mathbf{D} - \frac{1}{3}(\mathrm{tr}(2\mu_t \mathbf{D}))\boldsymbol{I_n}\right) - \frac{2}{3} < \rho > k\boldsymbol{I_n}$$

with $k = \frac{1}{2}\widetilde{\boldsymbol{V}'' \cdot \boldsymbol{V}''}$ the turbulent kinetic energy. By omitting the averaging notation, the Reynolds tensor $\boldsymbol{R}$ can be rewritten as:

$$\boldsymbol{\tau}^{turb} = \mu_t \left(2\mathbf{D} - \frac{2}{3}(\nabla \cdot \boldsymbol{V})\boldsymbol{I_n}\right) - \frac{2}{3}\rho k \boldsymbol{I_n}.$$

Turbulence modeling based on the Boussinesq assumption consists in determining the turbulent viscosity μ_t and the turbulent kinetic energy k. These quantities are described using 0, 1 or 2 transport equations.

The non linear eddy viscosity models rely on the same principles, the relationship between the Reynolds stress, the mean strain rate and vorticity tensors being more complicated which allows the Reynolds stress tensor to be non aligned with the strain rate. This is possible by introducing a tensor $\boldsymbol{a^{(ex)}}$ to define the Reynolds stress tensor $\boldsymbol{\tau}^{turb}$:

$$\boldsymbol{\tau}^{turb} = \mu_t \left(2\mathbf{D} - \frac{2}{3}\nabla \cdot \boldsymbol{V}\right) - \rho k \boldsymbol{I_n} - \rho k \boldsymbol{a^{(ex)}}.$$

These models are based on two-equation turbulence models.

The transport equations for the turbulent quantity have the form:

$$\partial_t(\rho\phi) + \boldsymbol{\nabla}(\rho \boldsymbol{V} \phi) = \boldsymbol{\nabla}(\mu_e^t \boldsymbol{\nabla}\phi) + S_t. \tag{2.19}$$

with ϕ the transported quantity, μ_e^t the turbulent effective viscosity and S_t a source term.

The turbulence models used in this thesis and their main characteristics are described below.

2.3.1 Spalart-Allmaras

The Spalart-Allmaras closure [Spalart 1992] is a one-equation closure calibrated on simple flows which is intensively used in aerodynamics. It provides satisfactory results on attached flows and gives a better description of velocity fields for detached flows than zero-equation models. Several versions and variants of this model, including curvature corrections, have been developed since the original version was drawn up. The implemented compressible and incompressible versions correspond to the standard Spalart-Allmaras model where the turbulent viscosity is a function of the variable $\tilde{\nu}$:

$$\mu_t = f_{v1}\rho \max(\tilde{\nu}, 0)$$

with $f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$, $\chi = \frac{\tilde{\nu}}{\mu/\rho}$ and C_{v1} a model constant defined below. Note that the effects of the turbulent kinetic energy k is neglected in this model.

The Spalart-Allamaras variable $\tilde{\nu}$ is transported using:

$$\partial_t(\rho\tilde{\nu}) + \boldsymbol{\nabla}(\rho\boldsymbol{V}\tilde{\nu}) = \boldsymbol{\nabla}(\mu_e^t\boldsymbol{\nabla}\tilde{\nu}) + S_t^{SA}. \tag{2.20}$$

with μ_e^t the turbulence effective viscosity defined as

$$\mu_e^t = \frac{\mu + \rho\tilde{\nu}}{\sigma_m} \tag{2.21}$$

with σ_m a model constant.

The components of the turbulence source flux are decomposed into P, D and d, respectively the production, destruction and diffusion terms:

$$S_t^{SA} = P - D + d$$

The production, destruction and diffusion terms are defined by:

$$P = c_{b1}\rho\tilde{\nu}\tilde{S}, \quad D = c_{w1}f_w\rho\left(\frac{\tilde{\nu}}{d_{wall}}\right)^2, \quad d = \frac{c_{b2}}{\sigma_m}\rho\|\boldsymbol{\nabla}\tilde{\nu}_t\|_2^2$$

with $\tilde{S} = \|\boldsymbol{w}\|_2 + \frac{\tilde{\nu}}{\kappa^2 d_{wall}^2}f_{v2}$ limited to be positive, d_{wall} the distance to the wall and the vorticity $\boldsymbol{w}$ defined as $\boldsymbol{w} = \boldsymbol{\nabla} \times \boldsymbol{V}$.

The missing model variables are set to:

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}$$

$$g = r + c_{w2}(r^6 - r), \quad r = \min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d_{wall}^2}, 10\right], \quad \chi = \frac{\tilde{\nu}}{\mu/\rho}$$

with the calibrated constants: $\kappa = 0.41$, $c_{b1} = 0.1355$, $\sigma_m = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$, $c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma_m}$.

The boundary conditions are set to:

$$\tilde{\nu}_\infty = 3\frac{\mu_\infty}{\rho_\infty},$$

$$\tilde{\nu}_{wall} = 0.$$

2.3.2 k-ε Launder Sherma

The k-ε Launder-Spalding [Launder 1974] is a two-equation turbulence closure. The aim of this closure is to predict the length scale and the time reversal of the largest turbulent eddies in the flow. To determine these characteristics, the k-ε model is based on solving the turbulent energy k and the dissipation rate ε. Its coefficients are calibrated on simple flows at equilibrium. The turbulent viscosity is a function of k and ε:

$$\mu_t = C_\mu f_\mu \rho\frac{k^2}{\varepsilon}$$

The baseline closure of k-ε Launder-Spalding is:

$$\begin{cases} \partial_t(\rho k) + \boldsymbol{\nabla}(\rho \boldsymbol{V} k) = \boldsymbol{\nabla}(\mu_e^{t(k)} \boldsymbol{\nabla} k) + S_t^k \\ \partial_t(\rho \varepsilon) + \boldsymbol{\nabla}(\rho \boldsymbol{V} \varepsilon) = \boldsymbol{\nabla}(\mu_e^{t(\varepsilon)} \boldsymbol{\nabla} \varepsilon) + S_t^\varepsilon \end{cases}, \tag{2.22}$$

with the respective turbulent effective viscosity defined as:

$$\begin{cases} \mu_e^{t(k)} = \left(\mu + \frac{\mu_t}{\sigma_k}\right) \\ \mu_e^{t(\varepsilon)} = \left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right) \end{cases}. \tag{2.23}$$

As for the Spalart-Allmaras model, the source term is decomposed in production, destruction and diffusion terms:

$$\begin{cases} S_t^k = P^k - D^k + d^k \\ S_t^\varepsilon = P^\varepsilon - D^\varepsilon + d^\varepsilon \end{cases} \tag{2.24}$$

These terms are defined by:

$$\begin{cases} P^k = \boldsymbol{\tau}_{ij}^{turb} \frac{\partial \boldsymbol{V}_i}{\partial x_j}, \\ P^\varepsilon = C_{\varepsilon_1} f_1 \frac{\varepsilon}{k} P^k \end{cases}, \quad \begin{cases} D^k = 2\,\rho\nu \left(\frac{\partial \sqrt{k}}{\partial y}\right)^2 \\ D^\varepsilon = 2\,\rho\nu\,\nu_t \left(\frac{\partial^2 u}{\partial y^2}\right)^2 \end{cases}, \quad \begin{cases} d^k = -\rho\varepsilon \\ d^\varepsilon = C_{\varepsilon_2} f_2 \frac{\varepsilon}{k} d^k \end{cases}. \tag{2.25}$$

And finally, the calibrated constants are set as $c_\mu = 0.09$, $\sigma_k = 1$, $\sigma_\varepsilon = 1.3$.

2.3.3 k-ω SST

As for the k-ε Launder-Spalding, the k-ω model is a two-equation turbulence closure that aims at predicting the length scale and the time reversal of the largest turbulent eddies in the flow. While the k-ε model is based on solving the turbulent energy k and the dissipation rate ε, the k-ω uses the turbulent energy k and the characteristic frequency ω of the largest eddies. It is well known that the simple k-ω closure is sensitive to boundary conditions, but can be integrated to the wall, while the k-ε closure is not sensitive to boundary conditions, but cannot be integrated to the wall unless using damping functions. By taking both advantages of the k-ε and k-ω closures, the k-ω-SST closure from Menter [Menter 1993] is deduced by mixing the k-ε closure far away from walls to k-ω closures close to the walls.

The turbulent viscosity is a function of k and ω:

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, \Omega f_2)},$$

with a_1, f_2 some constants defined below and Ω is the vorticity norm. The vorticity is defined by $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{V}$.

The baseline closure of k-ω-SST Menter is described by:

$$\begin{cases} \partial_t(\rho k) + \boldsymbol{\nabla}(\rho \boldsymbol{V} k) = \boldsymbol{\nabla}(\mu_e^{t(k)} \boldsymbol{\nabla} k) + S_t^k \\ \partial_t(\rho \omega) + \boldsymbol{\nabla}(\rho \boldsymbol{V} \omega) = \boldsymbol{\nabla}(\mu_e^{t(\omega)} \boldsymbol{\nabla} \omega) + S_t^\omega \end{cases}, \tag{2.26}$$

with the respective turbulent effective viscosity defined as:

$$\begin{cases} \mu_e^{t(k)} = (\mu + \sigma_k \mu_t) \\ \mu_e^{t(\omega)} = (\mu + \sigma_\omega \mu_t) \end{cases} . \tag{2.27}$$

The source term is decomposed in production, destruction and diffusion terms:

$$\begin{cases} S_t^k = P^k - D^k + d^k \\ S_t^\omega = P^\omega - D^\omega + d^\omega \end{cases} , \tag{2.28}$$

defined as

$$\begin{cases} P^k = \min(P, 20D^k), & D^k = \beta^* \rho \omega k & d^k = 0 \\ P^\varepsilon = \frac{\gamma}{\nu_t} P & D^\varepsilon = \beta \rho \omega^2 & d^\varepsilon = 2(1-f_1)(\frac{\rho \sigma_{w2}}{\omega})(\boldsymbol{\nabla} k \cdot \boldsymbol{\nabla} w) \end{cases} , \tag{2.29}$$

with

$$P = \boldsymbol{\tau}_{ij}^{turb} \frac{\partial \boldsymbol{V}_i}{\partial x_j}.$$

The missing constants are set to $\beta^* = 0.09, \kappa = 0.41, a_1 = 0.31$.

The other coefficients used in the previous equations are blended using the inner and outer values via the blending function:

$$\phi = f_1 \phi_1 + (1 - f_1)\phi_2,$$

with the blending function f_1 defined as:

$$\begin{cases} f_1 = \tanh\left(\arg_1^4\right) \\ f_2 = \tanh\left(\arg_2^2\right) \end{cases} . \tag{2.30}$$

$\arg_1$ and $\arg_2$ are defined such as:

$$\begin{cases} \arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega}\right), \frac{4\rho\sigma_{\omega 2} k}{\mathrm{CD}_{k\omega} d_{wall}^2}\right] \\ \arg_2 = \max\left(2\frac{\sqrt{k}}{\beta^* \omega d_{wall}}, \frac{500\nu}{d_{wall}^2 \omega}\right) \end{cases} . \tag{2.31}$$

with

$$\mathrm{CD}_{k\omega} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\boldsymbol{\nabla} k \cdot \boldsymbol{\nabla}\omega, 10^{-20}\right)$$

As example for the coefficient β:

$$\beta = f_1 \beta_1 + (1 - f_1)\beta_2.$$

And finally, the blended coefficients are:

$$\begin{cases} \sigma_{k1} = 0.85 \\ \sigma_{k2} = 1.0 \end{cases} , \begin{cases} \sigma_{\omega 1} = 0.5 \\ \sigma_{\omega 2} = 0.856 \end{cases} , \begin{cases} \beta_1 = 0.075 \\ \beta_2 = 0.0828 \end{cases} , \tag{2.32}$$

$$\begin{cases} \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \approx 0.553167 \\ \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} \approx 0.440355 \end{cases} . \tag{2.33}$$

The boundary conditions are set to:

$$\begin{cases} k_\infty = 9e^{-9}c_\infty^2 \\ \omega_\infty = 1e^{-6}\frac{\rho c_\infty^2}{\mu} \end{cases}, \tag{2.34}$$

$$\begin{cases} k_{wall} = 0 \\ \omega_{wall0} = 10\frac{6\nu}{\beta_1(\Delta_1 d_{wall})^2} \end{cases}. \tag{2.35}$$

2.3.4 k-ω EASM

Explicit and Implicit Algebraic Stress Models belong to the larger class of Algebraic Stress Models. They are based on an extension of the linear Boussinesq assumption, designed to add anisotropic behaviors to two-equation closures, corresponding to the previously described models in section 2.3. These models are non-linear and are built upon a full Reynolds Stress estimation, on the basis of a transport equation for a two-equation closure.

Contrary to two-equation closures, the Algebraic Stress Models can handle negative production of energy, rotational effects, etc. Furthermore, unsteady flows are characterized by a misalignment between the anisotropy tensor and the strain tensor [Carpy 2006]. As a consequence, two-equation closures are not able to predict these features, due to their linear nature, while Algebraic Stress Models can better predict such complex flows. Finally, an advantage of the formulations of Algebraic Stress Models is that they do not need to be calibrated.

The considered Explicit Algebraic Stress Model is based on the k-ω closure by Gatski & Speziale [Gatski 1993] with a near-wall formulation [Deng 1999]. As this model requires a complex decomposition of the Reynolds stress tensor, and that the aim of this work is not to derive new turbulence models, the reader is referred to the previously cited papers concerning the mathematical formulation of the model.

In turbulent flows including control mechanism, models of actuators are also needed. The following describes some of these models.

2.4 Actuation modeling - Synthetic jet

A synthetic jet is a fluidic actuator that injects momentum in the boundary layer by the mean of oscillatory blowing and suction phases. It has been found efficient for flow vectorization, mixing enhancement or detachment delay [Gilarranz 2002, Seifert 1996, Smith 1997]. This actuator is especially interesting for real-life flow control problems, because it is compact and does not require air supply, contrary to pulsating jets for instance.

As illustrated in the previous references, practical synthetic jets can be of different types. Nevertheless, they are usually composed by a cavity with a moving surface,

which generates inflow or outflow through a slot, as shown in Fig. (2.1). In practice, multiple technologies are known to develop such actuators and were the origin of several patents (US6056204, EP1040736, US 2010/0044459A1, US 2010/0045752A1, etc.). The principle of such actuators is to move a membrane or diaphragm up and down at high frequency to suck and blow the surrounding air. Examples of such technologies, are small membrane driven by piezoelectric actuators [Glezer 2002], regular speakers, pistons, etc. The numerical simulation of a synthetic jet in interaction with the flow in the outer domain is tedious, for several reasons. If one intends to represent exactly the device, the simulation of the flow in the deformable cavity [Rizzetta 1999, Holman 2005, Xia 2005, Utturkar 2002, Mittal 2001] should be achieved, which requires to use sophisticated methods like automated grid deformation, ALE (Arbitrary Lagrangian Eulerian) formulation, etc. Moreover, a significant part of the computational time could be devoted to the simulation of the flow inside the cavity, which is usually not the main purpose. Obviously, the introduction of the actuator makes the grid generation step more complex and the automatization of the process could be tedious, if possible. This is especially damageable in a design optimization framework, if one intends to optimize the actuator location, for instance. These reasons motivate the development and the use of simplified models in an industrial context. Some of them are detailed below.

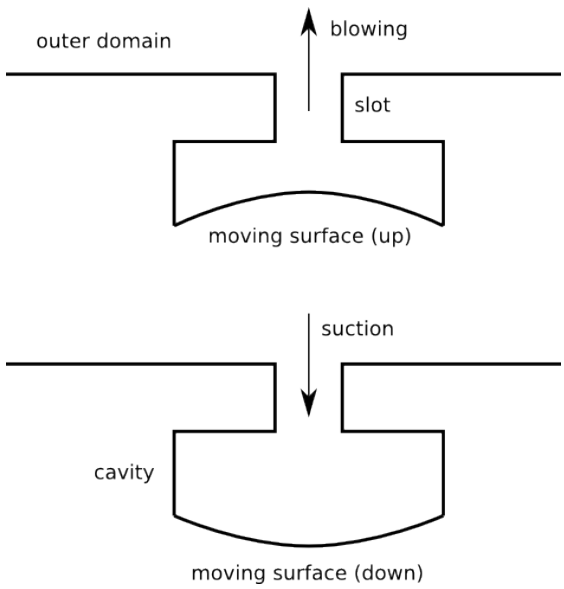


Figure 2.1: Synthetic jet principle.

2.4.1 Cavity model

A first simplification consists in using a fixed computational domain. In that case, the movement of the bottom surface of the cavity is modeled by imposing a prescribed boundary condition for the flow velocity [Dandois 2006] and possibly for the normal

pressure gradient [Kral 1997], as illustrated in Fig. (2.2). In this study, we impose the

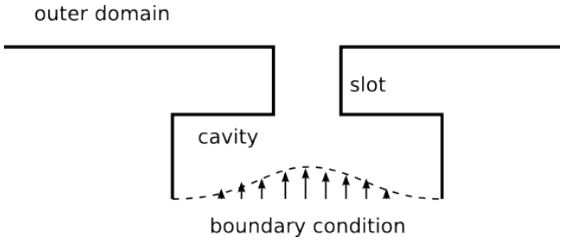


Figure 2.2: Cavity model.

value of the velocity at the bottom surface, as:

$$\boldsymbol{U} = U_c A_c(\xi_c) \sin(2\pi f t)\, \boldsymbol{\eta_c} \tag{2.36}$$

where U_c is the oscillation amplitude of the velocity at the cavity bottom surface, (ξ_c, η_c) a local frame system and f is the actuation frequency. $A_c(\xi_c)$ describes the velocity profile along the surface, which is here defined as:

$$A_c(\xi_c) = \sin^2\left(2\pi \frac{\xi_c - \xi_c^{min}}{\xi_c^{max} - \xi_c^{min}}\right) \tag{2.37}$$

with ξ_s^{min} and ξ_s^{max} the minimum and maximum bottom slot abscissae of the boundary considered.

Thus, the actuation is finally defined by the two parameters (U_c, f).

2.4.2 Slot model

The major drawback of the cavity model is related to the computational time used to simulate the flow inside the cavity, which is usually not negligible due to the necessity to use a refined grid in the cavity. Therefore, a subsequent simplification level consists in replacing the cavity by a simple boundary condition located at the slot extremity, as illustrated by Fig. (2.3). This shortcut reduces the computational burden, but does not allow to describe the flow in the cavity and the possible Helmholtz resonance phenomenon that can occur inside [Dandois 2007].

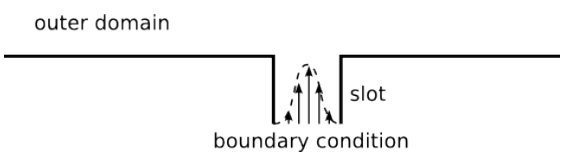


Figure 2.3: Slot model.

The boundary condition is formally equivalent to (2.37), but with different parameters:

$$\boldsymbol{U} = U_s A_s(\xi_s) \sin(2\pi f t)\, \boldsymbol{\eta_s} \tag{2.38}$$

$$A_s(\xi_s) = \sin^2\left(2\pi \frac{\xi_s - \xi_s^{min}}{\xi_s^{max} - \xi_s^{min}}\right) \tag{2.39}$$

The local frame (ξ_s, η_s) is now attached to the bottom slot surface. To impose an equivalent flow rate, the following relationship should be verified:

$$U_s = U_c \frac{\xi_c^{max} - \xi_c^{min}}{\xi_s^{max} - \xi_s^{min}} \tag{2.40}$$

2.4.3 Boundary condition model

Finally, the simplest model just consists in imposing a boundary condition for the velocity at the slot exit, as illustrated by Fig. (2.4). This model makes the grid generation process slightly easier, but the benefit is not significant, because the mesh should anyway account for the jet, as shown in the following numerical study. This model does not allow to describe the interaction between the flow in the outer domain and the flow exiting the slot. However, this model is certainly the most used in the literature [Rumsey 2006, Kral 1997, Vadillo 2002, Vadillo 2002, Pes 2002, Mallinson 1999, Capizzano 2005] due to its implementation ease.

Figure 2.4: Boundary model.

In practice, the boundary condition is the same as previously, defined by (2.38-2.39).

The practical numerical implementations of these models are further described in the chapters that follow.

Chapter 3

Numerical methods for simulation

Contents

3.1 Num3sis - a platform for numerical simulation 29
3.2 Mean turbulent flow 31
3.3 Turbulent equations 32
3.4 Space discretization 33
3.4.1 Convective fluxes 34
3.4.2 Viscous fluxes 39
3.4.3 Source terms 41
3.5 Time discretization 41
3.6 Boundary conditions 43
3.6.1 Weak approach for inlet, outlet and symmetry conditions 43
3.6.2 Strong approach for wall and jet condition 44
3.6.3 Strong iterative approach 45
3.6.4 Viscous boundary contribution 45

In this chapter, the numerical methods for solving the URANS equations are presented. Two different solvers have been used in this thesis to carry out the simulations. The thesis work has been focused on the development of an in-house compressible flow solver, which is part of the NUM3SIS simulation platform developed at INRIA. This chapter describes the platform and the numerical methods implemented to solve URANS equations. The second solver is an incompressible flow solver provided by the LHEEA (Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique) from CNRS UMR 6598 / ECN in France in the context of a collaborative work on flow control. This code has been used for a specific study on the BFS case and will be briefly described in the corresponding chapter.

3.1 Num3sis - a platform for numerical simulation

Most of the current industrial and research CFD codes are developed on the basis of a software architecture architecture coming from the 70s and 80s. This architecture is traditionnaly composed of a main program built with some static libraries for which the former describes the workflow and the latter contains the functions necessary for each

task in the workflow. This architecture has not really been questioned so far, although the complexity of the computations has grown significantly and that it appears clearly that it has several drawbacks.

First, the achievement of numerical experiments requires modularity, which allows to easily test different numerical methods (e.g. Roe flux v.s. HLLC flux), or modify physical parameters (e.g. inlet velocity profile). This ability to modify some parts of the code is critical to precisely benchmark numerical methods or models and easily prototype new methods. Obviously, traditional static architectures are not well suited to this purpose: usually, the implementation of a new method foo2 yields the addition of new functions and variables, that will juxtapose the existing ones foo0, foo1. Then, the code is always growing and becoming less and less readable.

Secondly, developments in CFD are now mainly collaborative projects, because it is increasingly difficult for one isolated person to master all modeling, numerical and computational aspects related to complex simulations. Unfortunately, a monolithic architecture is a real burden for collaborative development: since these codes are always growing (as explained above), the exchange of pieces of code and the maintenance is time consuming for long term and large scale projects. Typically, for each new method foo2, someone should verify if the proposed implementation is correct and will not generate conflicts with all other existing methods foo0, foo1 (for instance in case of addition of an argument).

Finally, the complexity of these codes makes the introduction of a new user tedious and time consuming: to modify a unique method, a new user should usually understand a large part of the code, even if it is not of interest for his study.

To overcome these limitations, the development of the NUM3SIS platform (http://num3sis.inria.fr) was initiated at INRIA. The software architecture is based on a new paradigm with a clear distinction between the platform and plugins.

The platform, written in C++ language, gathers all components dedicated to numerical simulation, that are considered as common to various disciplines. In practice, it consists of a set of abstractions, that can represent data or processes, commonly used in simulation like mesh, field, finite element, control cell, etc. As abstraction, numerical methods related to these objects are not implemented in the platform. On the contrary, a plugin contains a possible implementation of an abstraction defined in the platform. Note that a library written in a different language can be embedded into a plugin. In practice, plugins are dynamic libraries used by the platform at runtime.

This approach has several advantages: first, all the methods are not aggregated in a unique code, which improves readability and modularity. Then, it proposes a nice framework for collaborative development, because development of new plugins can be conducted independently from the platform or other plugins. Finally, an easy benchmarking procedure can be carried out by implementing the methods to be compared into different and independent plugins. One should underline that it is possible to change a parameter of a plugin, or the plugin itself, at runtime. This modifies the forthcoming

computation and enables the user to interact with the simulation.

During this thesis, all necessary developments required for the simulation of actuated flows have been implemented in the NUM3SIS platform. More specifically, I have extended the existing Eulerian code to solve unsteady turbulent RANS equations. These numerical methods are presented below. Although these methods are well known, a significant work has been achieved for their implementation in an object-oriented framework.

3.2 Mean turbulent flow

The equations solved for the turbulent compressible flow simulations are the Unsteady Favre-Averaged Navier-Stokes equations. We recall that for any variable x, with $\overline{x}$ the mean part and x' the fluctuations of x, the Reynolds average or filter is denoted by $x = \overline{x} + x'$ with $x' = 0$. The density weighted average, called Favre averaging, is denoted by $\widetilde{x} = \overline{\rho x}/\overline{\rho}$ with the related fluctuation part $x'' = x - \widetilde{x}$.

In conservative form, the averaged turbulent compressible equations can be written as (section 2.2.2):

$$\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) = \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) + \boldsymbol{\mathcal{S}}(\boldsymbol{W^c}) \tag{3.1}$$

with the averaged conservative variables $\boldsymbol{W^c} = (\overline{\rho}, \overline{\rho}\widetilde{u}, \overline{\rho}\widetilde{v}, \overline{\rho}\widetilde{w}, \overline{\rho}\widetilde{E})^T$, the inviscid flux $\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) = (\boldsymbol{F}(\boldsymbol{W^c}), \boldsymbol{G}(\boldsymbol{W^c}), \boldsymbol{H}(\boldsymbol{W^c}))^T$, the viscous flux $\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) = (\boldsymbol{R}(\boldsymbol{W^c}), \boldsymbol{S}(\boldsymbol{W^c}), \boldsymbol{T}(\boldsymbol{W^c}))^T$ and the source term $\boldsymbol{\mathcal{S}}(\boldsymbol{W^c})$.

The components of the inviscid flux in the global frame $\mathcal{R}_0(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{z}})$ are:

$$\boldsymbol{F}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{u} \\ \overline{\rho}\widetilde{u}^2 + \overline{p} \\ \overline{\rho}\widetilde{u}\widetilde{v} \\ \overline{\rho}\widetilde{u}\widetilde{w} \\ \widetilde{u}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{v} \\ \overline{\rho}\widetilde{v}\widetilde{u} \\ \overline{\rho}\widetilde{v}^2 + \overline{p} \\ \overline{\rho}\widetilde{v}\widetilde{w} \\ \widetilde{v}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}$$

$$\boldsymbol{H}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{w} \\ \overline{\rho}\widetilde{w}\widetilde{u} \\ \overline{\rho}\widetilde{w}\widetilde{v} \\ \overline{\rho}\widetilde{w}^2 + \overline{p} \\ \widetilde{w}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}$$

The components of the viscous flux are:

$$\boldsymbol{R}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{xx}} \\ \overline{\tau_{xy}} \\ \overline{\tau_{xz}} \\ \widetilde{u}\overline{\tau_{xx}} + \widetilde{v}\overline{\tau_{xy}} + \widetilde{w}\overline{\tau_{xz}} - \overline{q_x} \end{pmatrix}, \quad \boldsymbol{S}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{yx}} \\ \overline{\tau_{yy}} \\ \overline{\tau_{yz}} \\ \widetilde{u}\overline{\tau_{yx}} + \widetilde{v}\overline{\tau_{yy}} + \widetilde{w}\overline{\tau_{yz}} - \overline{q_y} \end{pmatrix}$$

$$\boldsymbol{T}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{zx}} \\ \overline{\tau_{zy}} \\ \overline{\tau_{zz}} \\ \widetilde{u}\overline{\tau_{zx}} + \widetilde{v}\overline{\tau_{zy}} + \widetilde{w}\overline{\tau_{zz}} - \overline{q_z} \end{pmatrix}$$

The averaged shear stress tensor $\overline{\boldsymbol{\tau}}$ is composed of a laminar and turbulent tensor: $\overline{\boldsymbol{\tau}} = \overline{\boldsymbol{\tau^{lam}}} + \overline{\boldsymbol{\tau^{turb}}}$. The laminar part is modeled considering Newtonian fluid:

$$\overline{\boldsymbol{\tau}^{lam}} = \overline{\mu}\left(\boldsymbol{\nabla}\widetilde{\boldsymbol{V}} + \boldsymbol{\nabla}\widetilde{\boldsymbol{V}}^T - \frac{2}{3}(\boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{V}})\boldsymbol{I_n}\right)$$

The turbulent part uses the Boussinesq eddy-viscosity assumption (section 2.3):

$$\overline{\boldsymbol{\tau}^{turb}} = -\overline{\rho\boldsymbol{V}''\otimes\boldsymbol{V}''} \approx \overline{\mu_t}\left(\boldsymbol{\nabla}\widetilde{\boldsymbol{V}} + \boldsymbol{\nabla}\widetilde{\boldsymbol{V}}^T - \frac{2}{3}(\boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{V}})\boldsymbol{I_n}\right) - \frac{2}{3}\overline{\rho}k\boldsymbol{I_n}$$

As for the shear stress tensor, the heat flux $\boldsymbol{q}$ is split in two parts: $\overline{\boldsymbol{q}} = \overline{\boldsymbol{q}^{lam}} + \overline{\boldsymbol{q}^{turb}}$. Both the laminar and turbulent parts are modeled by using a Fick or Fourier law:

$$\boldsymbol{q^{lam}}(\varepsilon,\kappa) = -\gamma\frac{\overline{\mu}}{\mathrm{Pr}}\boldsymbol{\nabla}\widetilde{\varepsilon}$$

$$\boldsymbol{q^{turb}}(\varepsilon,\kappa) = \overline{\rho\boldsymbol{V}''h''} \approx -\gamma\frac{\overline{\mu_t}}{\mathrm{Pr_t}}\boldsymbol{\nabla}\widetilde{\varepsilon}$$

with $\widetilde{\varepsilon} = \frac{\overline{p}}{(\gamma-1)\overline{\rho}}$ and the specific enthalpy $h = \mathrm{c_p}T$ where $\mathrm{c_p}$ is the specific coefficient at constant pressure and T the temperature.

Finally, to close the equations, the total energy $\widetilde{E} = \widetilde{\varepsilon} + \frac{1}{2}\widetilde{\boldsymbol{V}}\cdot\widetilde{\boldsymbol{V}} + k$, the adiabatic index is set to $\gamma = 1.4$, the Prandtl number is set to $\mathrm{Pr} = 0.72$, the turbulent Prandtl number is set to $\mathrm{Pr_t} = 0.9$ and the laminar viscosity μ is modeled by the Sutherland law, unless explicitly specified as being a constant.

For simplicity purpose, the average notations $\overline{.}$ and $\widetilde{.}$ are omitted in the following.

3.3 Turbulent equations

As for the mean equations, the turbulent equations in conservative form are written as:

$$\partial_t\boldsymbol{W^c_{(t)}} + \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{F}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) = \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{N}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) + \boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) \tag{3.2}$$

with in example for the Spalart-Allamars turbulence model $\boldsymbol{W^c_{(t)}} = (\rho\tilde{\nu})$, the inviscid turbulence flux $\boldsymbol{\mathcal{F}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) = (\rho\tilde{\nu}u, \rho\tilde{\nu}v, \rho\tilde{\nu}w)^T$, the viscous flux $\boldsymbol{\mathcal{N}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) = \mu_e^t\boldsymbol{\nabla}\tilde{\nu}$ with $\mu_e^t = \frac{\mu+\rho\tilde{\nu}}{\sigma_m}$, σ_m a model constant and the source flux $\boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}})$.

The components of the turbulence source flux in the global frame $\mathcal{R}_0(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{y}},\widehat{\boldsymbol{z}})$ are:

$$\boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c},\boldsymbol{W^c_{(t)}}) = \boldsymbol{P} - \boldsymbol{D} + \boldsymbol{d}$$

with P, D and d respectively the production, destruction and diffusion terms defined by the turbulence equations.

3.4 Space discretization

All the transport equations defined above are solved using Mixed finite-Element/finite-Volume (MEV) discretization with a vertex centered approach, developed by Dervieux and his collaborators at INRIA [Dervieux 1985, Debiez 2000, Desideri 1988]. Within the MEV framework, inviscid fluxes and source fluxes are discretized with a finite-volume approach while the viscous fluxes are discretized with a finite-element approach [Fezoui 1989, Francescatto 1998].

A polygonal bounded domain $\Omega \subset \mathbb{R}^n$ is considered with a boundary Γ, sub-divided into a tetrahedrization or triangulation $\mathcal{T}_h$ with elements T_i. Around each vertex s_i a finite-volume control cell C_i of a measure $m(C_i)$ is constructed. The finite-volume control cell is built around the node s_i by joining the barycenter of the triangles containing s_i and midpoint of the edges adjacent to s_i. The set of vertices which are joined to the vertex s_i is denoted by $\mathcal{N}(s_i)$. The subset of all the highest topological dimension polygons sharing the vertex s_i is denoted by $\mathcal{T}(s_i)$.

The inviscid and sources fluxes are computed on the dual control cells C_i while the viscous fluxes are computed on the elements T_i. A weak formulation of the Navier-Stokes equations can be expressed using a Galerkin approach.

By integrating the equation (3.1) over a control cell S_i (dual control cell or element) after multiplying by a test function φ_i, the weak formulation can be written as:

$$\int_{S_i} (\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c}))\varphi_i d\Omega = \int_{S_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}))\varphi_i d\Omega + \int_{S_i} (\boldsymbol{\mathcal{S}}(\boldsymbol{W^c}))\varphi_i d\Omega \tag{3.3}$$

The finite-volume method can be interpreted as a Galerkin method with the control cell $S_i = C_i$ and with the test function equals 1 inside the dual control cell and 0 outside. This test function, related to the control cell C_i is defined as:

$$\varphi_i^C(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \text{ is in } C_i \\ 0, & \text{else} \end{cases}$$

The variables $\boldsymbol{W^c}$ are considered to be constant on each control cells C_i, which is similar of having averaged states stored on each control cells, just as in the finite-volume methods. These constants are denoted by $\boldsymbol{W_i^c}$ on the cell C_i (see Fig. 3.4). The finite-element method is retrieved by using the control cell $S_i = T_i$. The polygon T_i is defined as a Lagrange $\mathcal{P}_1$ finite-element with the canonical basis functions denoted φ_i^T for each vertex s_i. Thus, φ_i^T is equal to unity at the vertex s_i and zero at the other vertices of the element. The $\mathcal{P}_1$ approximation of any function f on a polygon T_i can be obtained by:

$$f(\boldsymbol{X})_{|T_i} = \sum_{k \in T_i} f_k \varphi_k^T(\boldsymbol{X})$$

where f_i is the value of f at the vertex s_i.

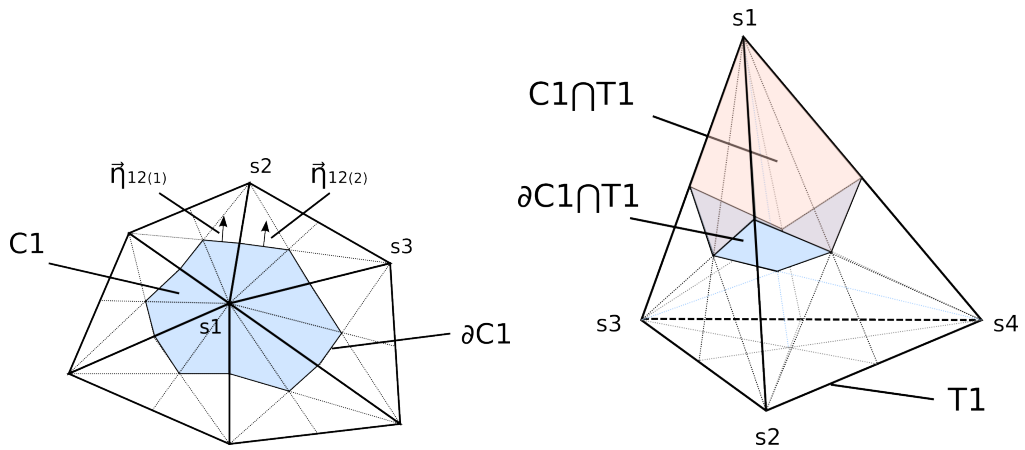


Figure 3.1: Illustration of a control cell in 2D and 3D.

The gradient for the function f on the finite elements are obtained using:

$$\boldsymbol{\nabla} f(\boldsymbol{X})_{|T_i} = \sum_{k \in T_i} f_k \boldsymbol{\nabla} \varphi_k^T(\boldsymbol{X})$$

From this element gradient, the nodal $\mathcal{P}_1$-Galerkin gradient $\boldsymbol{\nabla} f(\boldsymbol{X})_{|s_i}$ is computed as the weighted average of the element gradients $T \in \mathcal{T}(s_i)$ linked to the vertex s_i such as

$$\boldsymbol{\nabla} f(\boldsymbol{X})_{|s_i} = \frac{1}{m(C_i)} \sum_{T \in \mathcal{T}(s_i)} \frac{m(T)}{N_s(T)} \sum_{k \in T} f_k \boldsymbol{\nabla} \varphi_k^T(\boldsymbol{X}) \tag{3.4}$$

which can be expressed as

$$\boldsymbol{\nabla} f(\boldsymbol{X})_{|s_i} = \sum_{T \in \mathcal{T}(s_i)} \frac{\alpha_{T \cap C_i}}{N_s(T)} \sum_{k \in T} f_k \boldsymbol{\nabla} \varphi_k^T(\boldsymbol{X}) \tag{3.5}$$

by introducing the cell volume fraction $\alpha_{T \cap C_i} = \frac{m(T)}{m(C_i)}$ for the element T and $N_s(T)$ the number of vertices of the element T.

Finally, to reach a better accuracy within the boundary layer, control cells are constructed following Barth method [Barth 1995, Viozat 2001] by using the triangle circumcenter instead of barycenter for highly stretched elements. This approach gives regular instead of highly stretched control cells as shown on Fig. 3.4.

3.4.1 Convective fluxes

By using the Green-Ostrogradski theorem, the left term of 3.3 becomes:

$$\int_{C_i} (\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c})) \varphi_i^C d\Omega = m(C_i) \partial_t \boldsymbol{W_i^c} + \int_{\partial C_i} (\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) \cdot \widehat{\boldsymbol{\eta}})) d\sigma \tag{3.6}$$

with $\widehat{\boldsymbol{\eta}}$ the outward unit normal on ∂C_i, the boundary of the cell C_i.

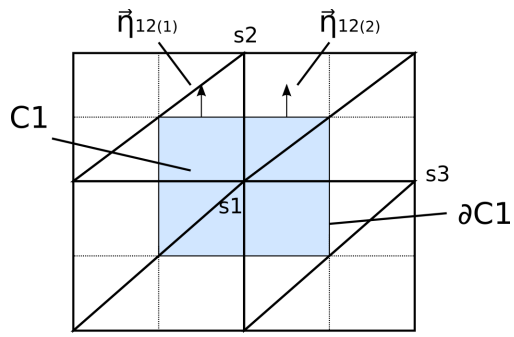


Figure 3.2: Illustration of a control cell in 2D using the Barth method.

Furthermore, as the domain is discretized,

$$\begin{aligned}\int_{\partial C_i}(\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})\cdot\widehat{\boldsymbol{\eta}}))d\sigma &= \sum_{s_j\in\mathcal{N}(s_i)}\int_{\partial C_i\cap\partial C_j}(\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})\cdot\widehat{\boldsymbol{\eta}}))d\sigma \\ &= \sum_{s_j\in\mathcal{N}(s_i)}\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{|ij}\cdot\int_{\partial C_i\cap\partial C_j}\widehat{\boldsymbol{\eta}}d\sigma \\ &= \sum_{s_j\in\mathcal{N}(s_i)}\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{|ij}\cdot\boldsymbol{\eta_{ij}}\end{aligned}$$

To obtain a high-order approximation in space, a MUSCL reconstruction technique is used [Van Leer 1979]. Any reconstructed primitive variables w^p_{ij} and w^p_{ji} from their initial state w^p_i and w^p_j at the common interface of the cells C_i and C_j read:

$$\begin{cases} w^p_{ij} = w^p_i + \frac{1}{2}\alpha_{ij}(\boldsymbol{\nabla}w^p)_{ij}\cdot\boldsymbol{IJ} \\ w^p_{ji} = w^p_j - \frac{1}{2}\alpha_{ji}(\boldsymbol{\nabla}w^p)_{ji}\cdot\boldsymbol{IJ}\end{cases}$$

The slopes $(\boldsymbol{\nabla}w^p)_{ij}\cdot\boldsymbol{IJ}$ and $(\boldsymbol{\nabla}w^p)_{ji}\cdot\boldsymbol{IJ}$ are approximated using the upwind nodal scheme [Camarri 2004]:

$$\Delta w^p_{ij} = \frac{1}{2}((1+\kappa)\Delta_{|C}w^p_{ij} + (1-\kappa)(2\Delta_{|N}w^p_i - \Delta_{|C}w^p_{ij}))$$

with the coefficient $\kappa = \frac{1}{3}$ that ensures to get the maximum accuracy for cartesian grids.

The nodal slope $\Delta_{|N}w^p_i$ is the projection of the nodal gradient on the edge IJ:

$$\Delta_{|N}w^p_i = \boldsymbol{\nabla}_{|s_i}w^p_i\cdot\boldsymbol{IJ}$$

And the centered slope $\Delta_{|C}w^p_{ij}$ is computed as:

$$\Delta_{|C}w^p_{ij} = w^p_j - w^p_i.$$

We denote by $\alpha_{ij}(\Delta_{|N}w^p_i, \Delta_{|C}w^p_{ij})$ the limiting coefficient. This limiter is ruled using a multidimensional limiting process (MLP) [Kim 2005, Yoon 2008] used in conjunction

with the Debiez limiter [Debiez 1998]:

$$\alpha_{ij}(\Delta_{|N}w_i^p, \Delta_{|C}w_{ij}^p)^{Debiez} = \begin{cases} 0 \quad \text{if } \Delta_{|N}w_i^p \Delta_{|C}w_{ij}^p < 0 \\ \frac{\min(|\Delta_{|N}w_i^p|, \min(\alpha_{MLP}|\Delta_{|C}w_{ij}^p|, \alpha_{MLP}|\Delta_{|N}w_i^p|))}{\Delta_{|N}w_i^p} \quad \text{else} \end{cases} \tag{3.7}$$

When the coefficient, $\alpha_{MLP} = 2$, the standard Debiez scheme is recovered. The MLP strategy is expressed as:

$$\alpha_{MLP} = \begin{cases} 2 & \text{if } |\Delta_{|N}w_i^p| \leq \varepsilon \text{ or } |\Delta_{|C}w_{ij}^p| \leq \varepsilon \\ \frac{2\min(|\Delta_{|max}w_i^p|, |\Delta_{|min}w_i^p|)}{\min(|\Delta_{|N}w_i^p|, |\Delta_{|C}w_{ij}^p|)} & \text{else} \end{cases} \tag{3.8}$$

with

$$\Delta_{|max}w_i^p = (\max_{k \in \mathcal{N}_i} w_k^p) - w_i^p$$

and with $\mathcal{N}_i$ the set of connected neighbors to the vertex s_i including the vertex itself. From our experience with Euler computations with shocks, the advantage of this limiting process is its robustness compared to classical TVD limiting approach. Moreover, it doesn't require any limiter freezing technique to ensure a good convergence of the computations to machine accuracy. Finally, the limiter is not necessary for subsonic flows and is typically set to $\alpha_{ij} = 1$ for such cases. If shocks occur during the transient state in the flow, using a limiter increases the robustness of the method.

The $\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{|ij}$ term is estimated using an approximate Riemann solver for which the associated numerical flux is $\boldsymbol{\Phi_{ij}} = \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})$. Riemann solvers have been designed to solve the Riemann problem, which consists in finding the solution of the unsteady Euler problem with an initial configuration where two constant states $\boldsymbol{W_L^p}$ and $\boldsymbol{W_R^p}$ are separated by a discontinuity or interface at $x = 0$. At $t = 0$, the interface is removed, and the solution pattern generated from the two initial states W_L and W_R typically consists of traveling rarefaction, contact discontinuity and shock waves. Such solution is shown on Fig. 3.3. Next to the rarefaction and shock waves, constant intermediate states W_L^* and W_R^* are generated. Approximate Riemann solvers consists in estimating waves speed and in providing an approximation of the W_L^* and W_R^* states. The rarefaction wave is replaced by a single traveling wave. The resulting formulation is condensed to provide a flux at an interface separating two states. Each facet in the computational domain is thus regarded as a discontinuity where a Riemann problem is solved, given a left state, a right state and a normal.

The resulting high-order numerical flux is simply computed using the extrapolated states $\boldsymbol{\Phi_{ij}}^{\text{High order}} = \boldsymbol{\Phi}(\boldsymbol{W_{ij}^c}, \boldsymbol{W_{ji}^c}, \boldsymbol{\eta_{ij}})$.

In the mean flow equations, the HLLC [Toro 1994] numerical flux is used for the internal domain and a modified version of the Steger-Warming flux [Steger 1981, Fezoui 1989] is used for boundaries.

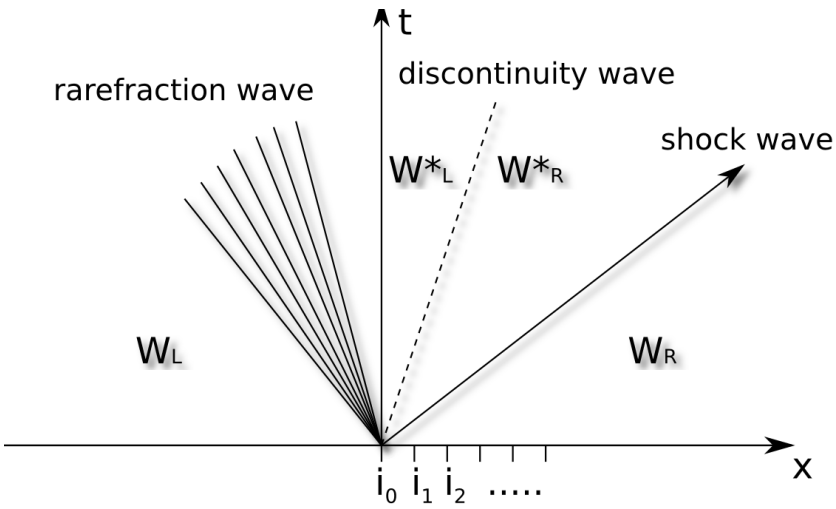


Figure 3.3: Typical solution of a Riemann problem, consisting in a rarefaction, discontinuity and shock wave. The initial configuration was composed of two separated left and right states, W_L and W_R.

3.4.1.1 HLLC flux

The HLLC flux approximates the rarefaction wave, contact discontinuity and shock waves into 3 distinct waves as on Fig. 3.4: a single left traveling wave at the speed S_L, a contact wave at the speed S_* and a right traveling wave at the speed S_R.

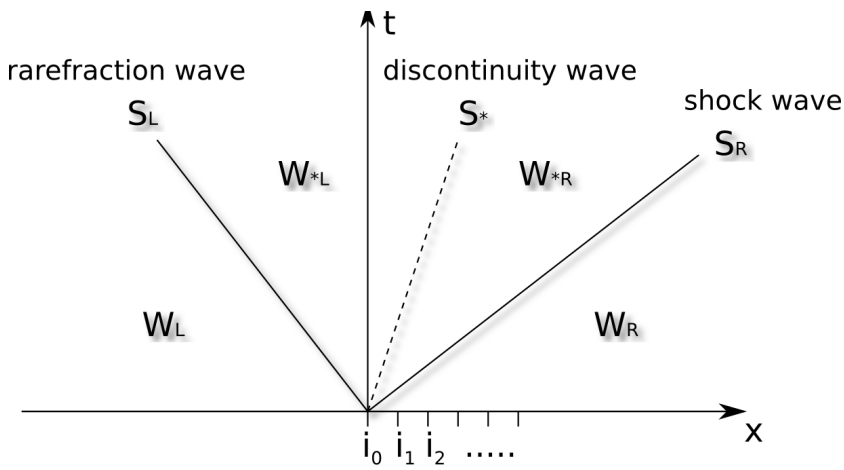


Figure 3.4: Riemann problem approximation and notation for the HLLC flux.

The HLLC numerical flux is defined as

$$\boldsymbol{\Phi_{ij}}^{HLLC} = \begin{cases} \boldsymbol{\Phi_L} & \text{if } 0 \leq S_L \\ \boldsymbol{\Phi_{*L}} & \text{if } S_L \leq 0 \leq S_* \\ \boldsymbol{\Phi_{*R}} & \text{if } S_* \leq 0 \leq S_R \\ \boldsymbol{\Phi_R} & \text{else} \end{cases} \tag{3.9}$$

$$\boldsymbol{\Phi_L} = \begin{pmatrix} \rho_L \boldsymbol{V}_L \cdot \boldsymbol{\eta_{ij}} \\ \boldsymbol{\Phi_L}[0]\boldsymbol{V}_L + p_L \boldsymbol{\eta_{ij}} \\ \boldsymbol{\Phi_L}[0](\varepsilon_L + \frac{p_L}{\rho_L}) \end{pmatrix} \tag{3.10}$$

All of the work to construct this flux relies on the estimation of the left fastest traveling wave speed S_L, the contact discontinuity wave speed S_* and the right fastest traveling wave speed S_R.

Let us introduce the ROE average [Roe 1981] for any variable w:

$$w^{ROE} = \frac{\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_L}{\sqrt{\rho_R} + \sqrt{\rho_R}}$$

The fastest signal velocities are computed as:

$$\begin{cases} S_L = \min(\boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij - c_L, \boldsymbol{V}^{ROE} \cdot \widehat{\boldsymbol{n}}ij - c^{AVG}) \\ S_R = \max(\boldsymbol{V}_R \cdot \widehat{\boldsymbol{n}}ij + c_R, \boldsymbol{V}^{ROE} \cdot \widehat{\boldsymbol{n}}ij + c^{AVG}) \end{cases} \tag{3.11}$$

with the average sound speed :

$$c^{AVG} = \sqrt{(\gamma - 1)(h^{ROE} - \frac{1}{2}(\boldsymbol{V}^{ROE} \cdot \boldsymbol{V}^{ROE}) - k^{ROE}) + \frac{2}{3}k^{ROE}} \tag{3.12}$$

The contact discontinuity speed is computed as:

$$S_* = \frac{p_R - p_L + \rho_L \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij(S_L - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij) - \rho_R \boldsymbol{V}_R \cdot \widehat{\boldsymbol{n}}ij(S_R - \boldsymbol{V}_R \cdot \widehat{\boldsymbol{n}}ij)}{\rho_L(S_L - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij) - \rho_R(S_R - \boldsymbol{V}_R \cdot \widehat{\boldsymbol{n}}ij)} \tag{3.13}$$

The star pressure is computed using:

$$p_* = p_L + \rho_L(S_L - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij)(S_* - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij); \tag{3.14}$$

From these, the star flux is computed as:

$$\boldsymbol{\Phi_{*L}} = \begin{cases} \rho_L \frac{S_L - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij}{S_L - S_*} S_* ||\boldsymbol{\eta_{ij}}|| \\ \boldsymbol{\Phi_{*L}}[0](S_* \widehat{\boldsymbol{n}}ij + (\boldsymbol{V_\tau})_L) + p_* \boldsymbol{\eta_{ij}} \\ \boldsymbol{\Phi_{*L}}[0](\varepsilon_L + \frac{p_L}{\rho_L} + S_L * (S_* - \boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij)) \end{cases} \tag{3.15}$$

with the tangent velocity defined as:

$$(\boldsymbol{V_\tau})_L = \boldsymbol{V}_L - (\boldsymbol{V}_L \cdot \widehat{\boldsymbol{n}}ij)\widehat{\boldsymbol{n}}ij \tag{3.16}$$

3.4.1.2 Modified Steger-Warming flux

The modified Steger-Warming flux [MacCormack 1989] is a non reflexive version of the original Steger-Warming flux [Steger 1981]. It is essentially used at boundaries instead of the HLLC flux to correctly filter the entering or exiting characteristics. The Steger-Warming flux is based on a flux splitting approach for which the flux is split into a positive flux due to waves moving to the right and a negative flux due to waves moving to the left of the considered face:

$$\boldsymbol{\Phi_{ij}}^{SW} = \boldsymbol{\mathcal{F}}i^{+}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}j^{-}(\boldsymbol{W_j^c}) \tag{3.17}$$

To be able to split this flux, the fluxes homogeneity property for Euler equations is used. This property is described as:

$$\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) = \boldsymbol{\mathcal{A}}(\boldsymbol{W^c})\boldsymbol{W^c} \tag{3.18}$$

with $\boldsymbol{\mathcal{A}}$ the inviscid flux Jacobian.

The Steger-Warming is built upon these two properties, which finally is expressed by:

$$\boldsymbol{\Phi_{ij}}^{SW} = \boldsymbol{\mathcal{F}}i^{+}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}j^{-}(\boldsymbol{W_j^c}) = \boldsymbol{\mathcal{A}}(\boldsymbol{W_i^c})^{+}\boldsymbol{W_i^c} + \boldsymbol{\mathcal{A}}(\boldsymbol{W_j^c})^{-}\boldsymbol{W_j^c} \tag{3.19}$$

The modified non-reflecting Steger-Warming flux alternative considers as incoming waves/information all the positive eigenvalues from the boundary. This is possible by modifying the incoming flux Jacobian $\boldsymbol{\mathcal{A}}(\boldsymbol{W_j^c})^{-}$ to use the interior state $\boldsymbol{W_i^c}$ such that:

$$\boldsymbol{\Phi_i}^{SW} = \boldsymbol{\mathcal{A}}(\boldsymbol{W_i^c})^{+}\boldsymbol{W_i^c} + \boldsymbol{\mathcal{A}}(\boldsymbol{W_i^c})^{-}\boldsymbol{W_e^c} \tag{3.20}$$

The details of the Jacobians are found in the appendix A.

3.4.1.3 Passive scalar flux

The passive scalars in the turbulence equations are transported reusing the density flux computed for the mean flow equations. This approach, introduced by Larrouturou, ensures positivity of the transported variables [Larrouturou 1989, Batten 1997a] and the corresponding flux is written as:

$$\boldsymbol{\Phi_{(t)ij}} = \boldsymbol{\Phi_{ij}}[0] \begin{cases} \boldsymbol{W^p_{(t)i}} & \text{if } \boldsymbol{\Phi_{ij}}[0] \geq 0 \\ \boldsymbol{W^p_{(t)j}} & \text{else} \end{cases} \tag{3.21}$$

3.4.2 Viscous fluxes

By integrating by part and using the Green-Ostrogradski theorem, the diffusive term of equation (3.3) becomes:

$$\int_{T_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}))\varphi_i^T d\Omega = -\int_{T_i} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla}\varphi_i^T d\Omega + \int_{\partial T_i} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n})\varphi_i^T d\sigma \tag{3.22}$$

In order to build the fluxes for the vertex s_i and match the finite volume framework, the integrals have to be rewritten at a vertex s_i by summation of all the terms that contributes to the vertex s_i. The diffusion term for the vertex s_i becomes:

$$\begin{aligned}\sum_{T_j, s_i \in T_j} (-\int_{T_j} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla} \varphi_i^{T_j} d\Omega + \int_{\partial T_j} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n}) \varphi_i^{T_j} d\sigma) = \\ \sum_{T_j, s_i \in T_j} -\int_{T_j} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla} \varphi_i^{T_j} d\Omega + \sum_{T_j, s_i \in T_j} \int_{\partial T_j} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n}) \varphi_i^{T_j} d\sigma\end{aligned} \tag{3.23}$$

As the physical variables $\boldsymbol{W^p}$ and the basis function φ_i^T are considered to be linear in the $\mathcal{P}_1$ finite element approximation on any tetrahedron T, the gradients of such variables and basis function gradient $\boldsymbol{\nabla} \varphi_i^T$ are constant by element.

Furthermore, in the viscous flux term $\boldsymbol{\mathcal{N}}$, all other unknown quantities are approximated on the element T following the finite element method. Their averaged value at the barycenter is directly used in $\boldsymbol{\mathcal{N}}(\boldsymbol{W^c})$. The averaging reads in this particular case:

$$\overline{f(\boldsymbol{X})}_{|T} = \frac{1}{N_s(T)} \sum_{k \in T} f_k.$$

The resulting viscous fluxes are then straightforward to evaluate:

$$\begin{aligned}\sum_{T_j, s_i \in T_j} -\int_{T_j} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla} \varphi_i^{T_j} d\Omega &= \sum_{T_j, s_i \in T_j} -\boldsymbol{\mathcal{N}}(T_j) \cdot \boldsymbol{\nabla} \varphi_i^{T_j} \int_{T_j} d\Omega \\ &= \sum_{T_j, s_i \in T_j} -\boldsymbol{\mathcal{N}}(T_j) \cdot \boldsymbol{\nabla} \varphi_i^{T_j} m(T_j)\end{aligned}$$

By denoting $\boldsymbol{\eta_{i,T_j}} = -\boldsymbol{\nabla} \varphi_i^{T_j} m(T_j)$, the viscous numerical flux is defined as:

$$\boldsymbol{\Upsilon_{T_j,i}} = \boldsymbol{\mathcal{N}}(T_j) \cdot \boldsymbol{\eta_{i,T_j}}$$

Thus, the equation (3.23) reads:

$$\sum_{T_j, s_i \in T_j} \int_{T_j} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c})) \varphi_i^{T_j} d\Omega = \sum_{T_j, s_i \in T_j} \boldsymbol{\Upsilon_{T_j,i}} + \sum_{T_j, s_i \in T_j} \int_{\partial T_j} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n}) \varphi_i^{T_j} d\sigma \tag{3.24}$$

The term $\int_{\partial T_j} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n}) \varphi_i^{T_j} d\sigma \approx \boldsymbol{\mathcal{N}}(\partial T_j) \cdot \boldsymbol{\eta_{\partial T_i}}$ concerns the boundary conditions contribution and is derived depending on the considered type of boundary condition.

3.4.3 Source terms

The source terms are discretized in a finite volume way, by considering the variables are constant on the control cell C_i. The source term of 3.3 becomes:

$$\int_{C_i} (\boldsymbol{\mathcal{S}}(\boldsymbol{W^c}))\varphi_i^C d\Omega = m(C_i)\boldsymbol{\mathcal{S}}(\boldsymbol{W_i^c})$$

If the source terms require gradient values, a weighted average of the $\mathcal{P}_1$-Galerkin gradients is used, as for the inviscid fluxes MUSCL reconstruction.

3.5 Time discretization

An implicit second-order time discretization is obtained by using a dual time step approach [Jameson 1991] and a backward time integration. By denoting $\delta_1^2\Lambda = \Lambda^2 - \Lambda^1$ the variation of the variable Λ from the state 1 to the state 2, we obtain as second-order implicit time integration scheme:

$$m(C_i)\frac{3\boldsymbol{W_i^c}^{(n+1)} - 4\boldsymbol{W_i^c}^{(n)} + \boldsymbol{W_i^c}^{(n-1)}}{2\delta_n^{n+1}t} + \boldsymbol{\mathcal{R}_i}(\boldsymbol{W^c}^{(n+1)}) = 0 \tag{3.25}$$

with the computed residual $\boldsymbol{\mathcal{R}_i}$ for the control cell C_i.

To solve this problem, we introduce a sub-iteration state of index k, such as:

$$\boldsymbol{W_i^c}^{n+1} = \lim_{k\to\infty} \boldsymbol{W_i^c}^{(k+1)}$$

The linearization of equation (3.25) around the state of index k, with a local time step $\delta_k^{k+1}t_i$ yields:

$$\begin{aligned}\left(\left(\frac{m(C_i)}{\delta_k^{k+1}t_i} + \frac{3m(C_i)}{2\delta_n^{n+1}t}\right)\boldsymbol{I_n} + \boldsymbol{\mathcal{J}_i^*}(\boldsymbol{W^c}^{(k)})\right)\delta_k^{k+1}\boldsymbol{W_i^c} = \\ -\boldsymbol{\mathcal{R}_i}(\boldsymbol{W^c}^{(k)}) + m(C_i)\frac{\delta_{n-1}^n\boldsymbol{W_i^c} - 3\delta_n^k\boldsymbol{W_i^c}}{2\delta_n^{n+1}t}\end{aligned} \tag{3.26}$$

with $\boldsymbol{\mathcal{J}_i^*}(\boldsymbol{W^c}^{(k)})$ an approximate Jacobian of the numerical residuals.

The approximation of the Jacobian does not change the converged solution. It only affects the rate of convergence of the inner or sub-iterations k towards a stationary state. Any approximation is acceptable as long as the implicit operator at the left hand side of the equation can be inverted. Unfortunately, one has to set the balance between rate of convergence and the easiness of inverting the Jacobian. In particular, the exact Jacobian leads to a Newton method but the inversion of such matrix is extremely hard even using the most advanced iterative solvers.

The construction of approximate Jacobian consists of three parts arising from the inviscid fluxes, the viscous terms and the source terms. All of these terms are taken independently to build the full matrix.

The Jacobian of the inviscid fluxes is based on the linearization of a first-order Rusanov flux [Rusanov 1961] for the inner state and of a first-order modified Steger-Warming flux at the boundaries. The Rusanov flux is extremely dissipative and allows to increase the robustness of matrix inversion as it adds diagonal dominance. The use of the Rusanov flux, or spectral radius Jacobian approximation, coupled with a Jacobi or Symmetric Gauss Seidel iterative solver, is usually the base of matrix-free approaches [Kloczko 2006, Dwight 2006].

The Jacobian of the viscous terms is built on an exact linearization, as in [Fezoui 1989].

For the source terms, arising for the turbulence variables, the full source fluxes Jacobian for Spalart-Allmaras is computed and only the positive contribution to the diagonal are kept for k-ω SST closure.

The resulting assembled system for the mean flow variables is rewritten as:

$$\boldsymbol{\mathcal{A}}^*(\boldsymbol{W}^{\boldsymbol{c}(k)})\delta_k^{k+1}\boldsymbol{W}^{\boldsymbol{c}} = -\boldsymbol{\mathcal{R}}^*(\boldsymbol{W}^{\boldsymbol{c}(k)}) \tag{3.27}$$

with $\boldsymbol{\mathcal{A}}^*(\boldsymbol{W}^{\boldsymbol{c}(k)})$ the approximated Jacobian matrix including the dual time stepping and $\boldsymbol{\mathcal{R}}^*$ the full residual of the mean flow. This system is solved for each sub-iteration k with a Symmetric Gauss-Seidel (SGS) iterative algorithm. The use of a delta form for the implicit formulation allows to employ an approximated Jacobian without loosing the second order accuracy in space reached with the MUSCL extrapolation.

In practice, the time integration of the mean flow equations and the turbulence equations are carried out using a weak coupling. The turbulence variables are frozen to build the system for mean flow variables, whereas the system for the turbulence variables is built with frozen mean flow variables. As a result, at each sub-iterations k, two linear systems are solved:

$$\begin{cases} \boldsymbol{\mathcal{A}}^*(\boldsymbol{W}^{\boldsymbol{c}(k)})\delta_k^{k+1}\boldsymbol{W}^{\boldsymbol{c}} = -\boldsymbol{\mathcal{R}}^*(\boldsymbol{W}^{\boldsymbol{c}(k)}) \\ \boldsymbol{\mathcal{A}}^*_{(\boldsymbol{t})}(\boldsymbol{W}^{\boldsymbol{c}(k)}, \boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})}{}^{(k)})\delta_k^{k+1}\boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})} = -\boldsymbol{\mathcal{R}}^*_{(\boldsymbol{t})}(\boldsymbol{W}^{\boldsymbol{c}(k)}, \boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})}{}^{(k)}) \end{cases} \tag{3.28}$$

with $\boldsymbol{\mathcal{A}}^*_{(\boldsymbol{t})}(\boldsymbol{W}^{\boldsymbol{c}(k)}, \boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})}{}^{(k)})$ the approximated Jacobian matrix and $\boldsymbol{\mathcal{R}}^*_{(\boldsymbol{t})}$ the full residual for the turbulent system.

The states are then updated using:

$$\begin{cases} \boldsymbol{W}^{\boldsymbol{c}k+1} = \boldsymbol{W}^{\boldsymbol{c}k} + \delta_k^{k+1} \\ \boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})}{}^{k+1} = \boldsymbol{W}^{\boldsymbol{c}}_{(\boldsymbol{t})}{}^{k} + \delta_k^{k+1} \end{cases} \tag{3.29}$$

3.6 Boundary conditions

Within the implicit MEV framework, multiple possibilities are offered to impose boundary conditions. The strong boundary condition consists in setting the value of the solution at the boundary. On the counterpart, weak boundary conditions consist of imposing the boundary fluxes. The following presents these two forms and their use in the current framework.

3.6.1 Weak approach for inlet, outlet and symmetry conditions

In the weak form, the boundary flux is estimated using a fictive state $\boldsymbol{W_b^c}$ and the interior state $\boldsymbol{W_i^c}$. Depending on the boundary condition used, the fictive state $\boldsymbol{W_b^c}$ can be a function of the interior $\boldsymbol{W_i^c}$ and/or the exterior $\boldsymbol{W_o^c}$ states: $\boldsymbol{W_b^c} = \boldsymbol{W_b^c}(\boldsymbol{W_i^c}, \boldsymbol{W_o^c})$. The resulting flux using the boundary and interior state can be computed with any numerical flux (independently from the interior domain) as illustrated in the Fig. 3.5.

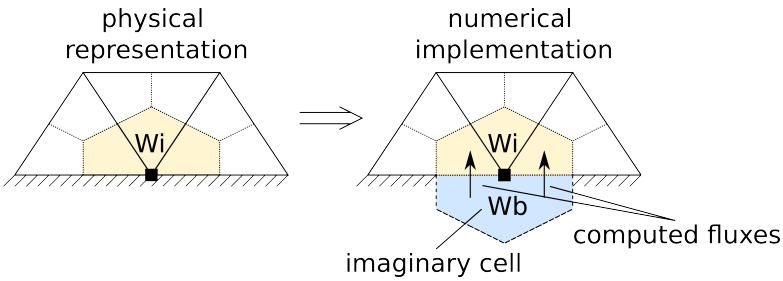


Figure 3.5: Illustration of the implementation of weak boundary conditions.

The previously described modified Steger-Warming flux is preferred over the interior flux for all boundary conditions, as it filters the incoming characteristics from the outgoing ones.

The computation of the fictive state is carried out using imposed density and velocity with extrapolation from the interior of the pressure for subsonic inlets, imposed pressure with extrapolation of other state variables for subsonic outlets, and symmetry conditions. Multiple possibilities can be used for the extrapolation. The simplest one, using the boundary variable itself is selected here as an example with $\boldsymbol{W_b^c} = \boldsymbol{W_i^c}$.

In particular, under assumption of local isentropic flow, exterior states for the freestream boundary conditions are computed using the Riemann invariants [Hirsch 1990]:

$$\boldsymbol{W^R} = \begin{pmatrix} \mathcal{R}^- \\ S \\ \boldsymbol{V_{\widehat{\tau}}} \\ \mathcal{R}^+ \end{pmatrix} = \begin{pmatrix} \boldsymbol{V_{\widehat{n}}} - \frac{c}{\gamma-1} \\ \mathrm{c_v} \log(\frac{p}{\rho^\gamma}) \\ \boldsymbol{V_{\widehat{\tau}}} \\ \boldsymbol{V_{\widehat{n}}} + \frac{c}{\gamma-1} \end{pmatrix} \tag{3.30}$$

For an inlet, the Riemann invariants coming into the domain, associated with the positive eigen values (S, $\boldsymbol{V}_{\widehat{\boldsymbol{\tau}}}$, and $\boldsymbol{V}_{\widehat{\boldsymbol{\tau}}}$) are set to be equals to the infinite Riemann invariants $\boldsymbol{W_o^R}$. The remaining terms can be extrapolated from the interior of the domain.

This leads to for an inlet:

$$\boldsymbol{W_b^R} = \begin{pmatrix} \mathcal{R}_i^- \\ S_o \\ \boldsymbol{V}_{\widehat{\boldsymbol{\tau}}o} \\ \mathcal{R}_o^+ \end{pmatrix} \tag{3.31}$$

For an outlet, only the Riemann invariant associated with the negative eigen value is computed from the exterior of the domain:

$$\boldsymbol{W_b^R} = \begin{pmatrix} \mathcal{R}_o^- \\ S_i \\ \boldsymbol{V}_{\widehat{\boldsymbol{\tau}}i} \\ \mathcal{R}_i^+ \end{pmatrix} \tag{3.32}$$

For symmetry boundary conditions, the boundary state is taken as a symmetry of the interior state with respect to the boundary unit normal $\widehat{\boldsymbol{n}}_b$:

$$\boldsymbol{W_b^p} = \begin{pmatrix} \rho_i \\ \boldsymbol{V}_i - 2(\boldsymbol{V}_i\widehat{\boldsymbol{n}}_b)\widehat{\boldsymbol{n}}_b \\ p_i \end{pmatrix} \tag{3.33}$$

3.6.2 Strong approach for wall and jet condition

The strong boundary condition or dirichlet approach consists in imposing the solution at the boundary. A way to implement it is to modify the Jacobian and the residuals in such a way that the boundary condition is verified implicitly when solving the linear system. This boundary condition is used when the value of one particular conservative variable is completely defined at the boundary. Wall boundary conditions are imposed by setting at the boundary a zero momentum for the mean flow equations and appropriate turbulent variables for the turbulence equations. Jet boundary conditions are as well imposed with this methodology.

For adiabatic walls or other non time varying boundary conditions, the target is to retrieve $\delta_k^{k+1}\boldsymbol{W_i^c} = 0$ at boundary during the system resolution. To do so, the residuals are corrected by imposing a zero residual and identity lines in the flux Jacobian matrix,

at the corresponding momentum lines:

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{i}}^{*} = \begin{pmatrix} \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \dots & X & X & X & X & X & \dots \\ \dots & 0 & 1 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 1 & 0 & \dots \\ \dots & X & X & X & X & X & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \end{pmatrix}, \quad \boldsymbol{\mathcal{R}}_{\boldsymbol{i}}^{*} = \begin{pmatrix} \vdots \\ X \\ 0 \\ 0 \\ 0 \\ X \\ \vdots \end{pmatrix} \tag{3.34}$$

The same modifications hold for the jet boundary condition, with an additional extra diagonal matrix update to impose the velocity value. The Jacobian and residuals correction follows:

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{i}}^{*} = \begin{pmatrix} \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \dots & X & X & X & X & X & \dots \\ \dots & -u_i^{jet} & 1 & 0 & 0 & 0 & \dots \\ \dots & -v_i^{jet} & 0 & 1 & 0 & 0 & \dots \\ \dots & -w_i^{jet} & 0 & 0 & 1 & 0 & \dots \\ \dots & X & X & X & X & X & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \end{pmatrix}, \quad \boldsymbol{\mathcal{R}}_{\boldsymbol{i}}^{*} = \begin{pmatrix} \vdots \\ X \\ 0 \\ 0 \\ 0 \\ X \\ \vdots \end{pmatrix} \tag{3.35}$$

with u_i^{jet}, v_i^{jet} and w_i^{jet} the jet velocity components defined at cell i. In practice, the jet velocity is computed as in section 2.4.

3.6.3 Strong iterative approach

In cases where it is cumbersome to modify the linear system to impose implicitly the value of the solution, an iterative technique can replace the implicit strategy. This can occur for instance when the user would like to impose a primitive variable while the Jacobian and residuals are expressed using conservative variables. It is thus possible to use the dual time step approach to implement iteratively the boundary condition, in order to reach a target $\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}target}$ at the boundary at the step $n+1$.

In other words, this boundary condition consists in imposing at each dual time step sub-iterations the targeted value of the variable. As described in [Ashcroft 2004], the internal fields and the boundary conditions converge towards the solution with the targeted boundary condition at the next physical step.

3.6.4 Viscous boundary contribution

The viscous boundary term in 3.24 has to be evaluated. The following describes specific boundary conditions to evaluate the term $\int_{\partial T_j} (\boldsymbol{\mathcal{N}}(\boldsymbol{W}^{\boldsymbol{c}}) \cdot \boldsymbol{n}) \varphi_i^{T_j} d\sigma \approx \boldsymbol{\mathcal{N}}(\partial T_j) \cdot \boldsymbol{\eta}_{\boldsymbol{\partial T_i}}$.

3.6.4.1 Wall condition

For walls, the following holds:

$$\boldsymbol{\mathcal{N}}(\boldsymbol{W}^{\boldsymbol{c}}) \cdot \boldsymbol{\eta_i} = \begin{pmatrix} 0 \\ \boldsymbol{\tau}_{xx}\eta_{i_x} + \boldsymbol{\tau}_{xy}\eta_{i_y} + \boldsymbol{\tau}_{xz}\eta_{i_z} \\ \boldsymbol{\tau}_{yx}\eta_{i_x} + \boldsymbol{\tau}_{yy}\eta_{i_y} + \boldsymbol{\tau}_{yz}\eta_{i_z} \\ \boldsymbol{\tau}_{zx}\eta_{i_x} + \boldsymbol{\tau}_{zy}\eta_{i_y} + \boldsymbol{\tau}_{zz}\eta_{i_z} \\ -(q_x\eta_{i_x} + q_y\eta_{i_y} + q_z\eta_{i_z}) \end{pmatrix} \tag{3.36}$$

As the wall conditions are strongly enforced at the boundaries, the viscous boundary fluxes corresponding to the momentum equations are not necessary to be computed. The last term $q_x\eta_{i_x} + q_y\eta_{i_y} + q_z\eta_{i_z} = 0$ in case of adiabatic walls. In the considered cases, there is thus no viscous contribution of wall boundary conditions as the walls are considered to be adiabatic.

3.6.4.2 Symmetry condition

Using the Einstein summation convention for the ease of manipulation, the viscous force on a plane surface is:

$$\begin{aligned} \boldsymbol{\tau}_{ij}\widehat{\boldsymbol{n}}_j &= (\mu + \mu_t)\left(\frac{\partial \boldsymbol{V}_i}{\partial x_j} + \frac{\partial \boldsymbol{V}_j}{\partial x_i}\right)\widehat{\boldsymbol{n}}_j - \frac{2}{3}(\mu + \mu_t)\frac{\partial \boldsymbol{V}_k}{\partial x_k}\delta_{ij}\widehat{\boldsymbol{n}}_j \\ &= (\mu + \mu_t)\left(\widehat{\boldsymbol{n}}_j\frac{\partial \boldsymbol{V}_i}{\partial x_j} + \frac{\partial \boldsymbol{V}_j\widehat{\boldsymbol{n}}_j}{\partial x_i}\right) - \frac{2}{3}(\mu + \mu_t)\frac{\partial \boldsymbol{V}_k}{\partial x_k}\widehat{\boldsymbol{n}}_i \end{aligned} \tag{3.37}$$

The symmetry boundary conditions are determined considering that the normal velocity to the surface and normal gradients for all variables are zero. These two conditions are written:

$$\boldsymbol{V}_j\widehat{\boldsymbol{n}}_j = 0 \quad \text{and} \quad \widehat{\boldsymbol{n}}_j\frac{\partial}{\partial x_j} = 0 \tag{3.38}$$

Thus, with these assumptions:

$$\boldsymbol{\tau}_{ij}\widehat{\boldsymbol{n}}_j = -\frac{2}{3}(\mu + \mu_t)\frac{\partial \boldsymbol{V}_k}{\partial x_k}\widehat{\boldsymbol{n}}_i \tag{3.39}$$

The viscous and heat flux for the energy equation, are then:

$$\begin{aligned} \boldsymbol{V}_i\boldsymbol{\tau}_{ij}\widehat{\boldsymbol{n}}_j - \boldsymbol{q}_j\widehat{\boldsymbol{n}}_j &= -\frac{2}{3}(\mu + \mu_t)\frac{\partial \boldsymbol{V}_k}{\partial x_k}\boldsymbol{V}_i\widehat{\boldsymbol{n}}_i - \left(\gamma\frac{\overline{\mu}}{\mathrm{Pr}} + \gamma\frac{\overline{\mu_t}}{\mathrm{Pr_t}}\right)\widehat{\boldsymbol{n}}_j\frac{\partial \varepsilon}{\partial x_j} \\ &= 0 \end{aligned} \tag{3.40}$$

Finally, the viscous flux for an adiabatic symmetry condition is written as:

$$\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\eta_i} = \begin{pmatrix} 0 \\ -\frac{2}{3}(\mu + \mu_t)(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})\eta_{i_x} \\ -\frac{2}{3}(\mu + \mu_t)(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})\eta_{i_y} \\ -\frac{2}{3}(\mu + \mu_t)(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})\eta_{i_z} \\ 0 \end{pmatrix} \tag{3.41}$$

3.6.4.3 Other boundary conditions

For all other boundary conditions, the full viscous term $\boldsymbol{\mathcal{N}}(\partial T_j) \cdot \boldsymbol{\eta_{\partial T_i}}$ is computed.

Chapter 4

Validation for uncontrolled flows

Contents

4.1 Turbulent subsonic flow over a flat plate 49

4.2 Turbulent subsonic flow around NACA0012 airfoil 51

This chapter details some validation exercises of numerical methods implemented in NUM3SIS on relevant test cases. These test cases are mainly subsonic flow as they represent the flow regime targeted in the last part about optimization. For all the test cases, the results are compared to FUN3D and/or CFL3D compressible solvers from the NASA Langley Research Center, and possibly to experimental databases. These data are available at the turbulence modeling resource website http://turbmodels.larc.nasa.gov.

4.1 Turbulent subsonic flow over a flat plate

It is a steady test case where a turbulent boundary layer is developing along a flat plate. The configuration and flow boundary conditions are represented on Fig. 4.1. To

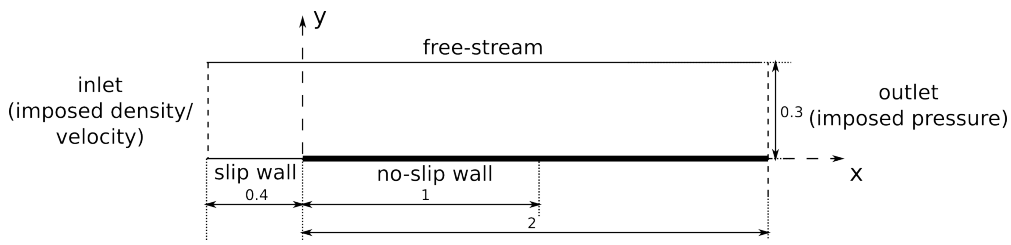


Figure 4.1: Flat plate test case configuration.

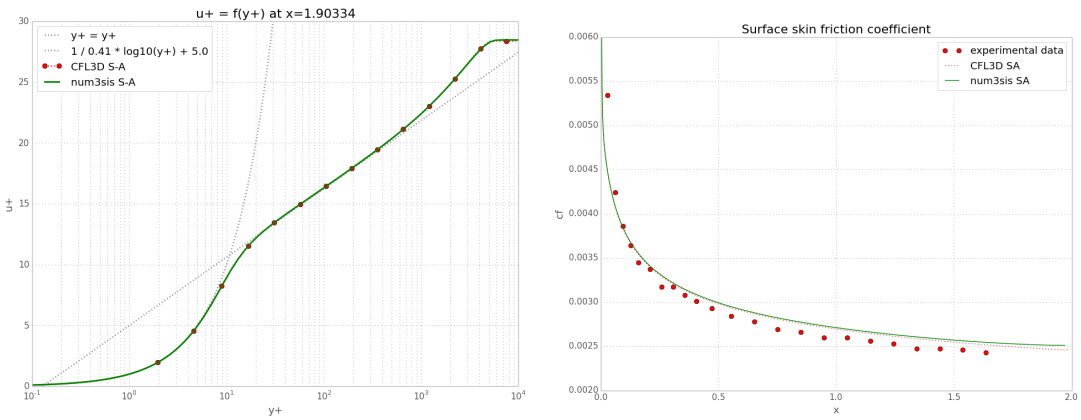
match a mach number $M = 0.2$, a Reynolds number of 5.10^5 based on the length 1 (half the size of the plate), the parameters given in the Tab. 4.1 are chosen. The inlet flow is considered as a subsonic boundary condition, where the density and velocity only are imposed. The outlet is considered as a subsonic outlet with an imposed pressure. the free-stream boundary state at the top of the domain is determined using Riemann invariants. The turbulence models used are both the Spalart-Allmaras and k-ω SST models. The mesh contains 36100 finite volume cells and the maximal triangle aspect ratio in the domain is approximately 25700. The residuals are reduced of 7 orders of

Variable	Value
ρ_∞	1.363193
$\boldsymbol{V}_\infty$	69.437742
p_∞	115056
μ_∞	1.9e-05

Table 4.1: Flat plate test case flow parameters.

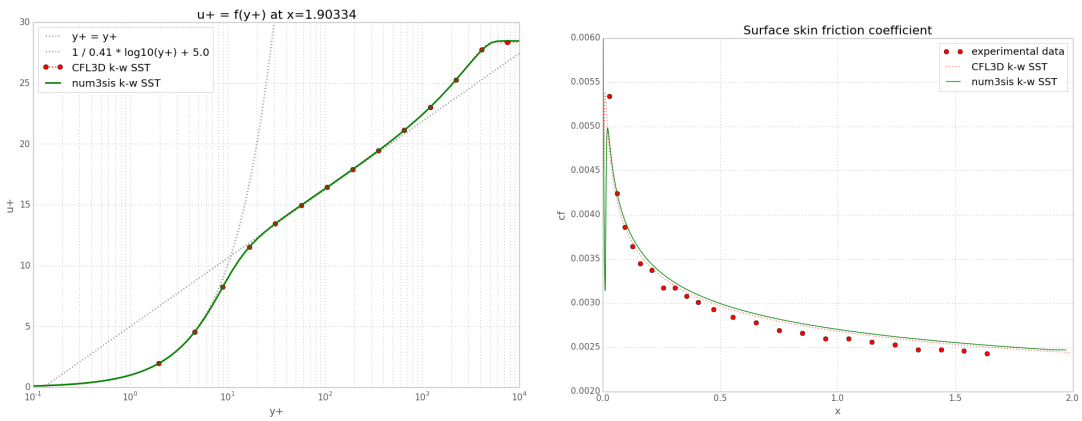
magnitude.

The computed flow is compared to the experimental data from Wieghardt and Tillman [Wieghardt 1951], reported by Coles and Hirst [Coles 1968], to the theoretical laws describing the viscous sublayer and log-law region layer and to computational results from CFL3D code. The non-dimensional velocity profile $u^+ = \frac{u}{u_\tau}$ in the near wall region with respect to the non-dimensional velocity $y^+ = \frac{y\,u_\tau}{\nu}$, with $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ and τ_w the wall shear stress, is shown on the Fig. 4.2(a). As seen, the computed velocity profiles matches CFL3D solution on a their finest finest 545×385 grid and theoretical ones. Surface skin friction coefficient results are good compared to experimental data using both turbulence models as shown in Fig. 4.2(b). The maximal non dimensional velocity y^+ is kept under 0.4 for both cases with the first point to the wall located at a distance of 2.10^{-6}.



(a) (b)

Figure 4.2: Flat plate profile at $x = 1.90334$ (a) and friction coefficient with the S-A turbulence model (b).



(a) (b)

Figure 4.3: Flat plate profile at $x = 1.90334$ (a) and friction coefficient with the k-ω-SST turbulence model (b). The friction coefficient located at the leading edge shows a breakdown noticed using CFL3D as well.

4.2 Turbulent subsonic flow around NACA0012 airfoil

For this test case, the configuration at an incidence of 10 deg and 15 deg are chosen and flow conditions are similar to the case presented by the NASA Langley Research Center (http://turbmodels.larc.nasa.gov/naca0012_val.html). The NACA0012 used in this validation is the altered formulation, allowing the airfoil to close at chord= 1 with a sharp trailing edge. The shape is described by equation:

$$\begin{aligned} y = \pm\, 0.594689181(0.298222773\sqrt{x} - \\ 0.127125232x - 0.357907906x^2 + 0.291984971x^3 - 0.105174696x^4) \end{aligned} \tag{4.1}$$

The airfoil is then rotated to match the given incidence. The infinite values are considered as free-stream boundary conditions using Riemann invariants. The O-grid used for the 10 deg computation is represented in the Fig. 4.4. The 15 deg is similar. They contains 48512 points for the 10 deg case and 44870 points for the 15 deg case. They are clustered around the airfoil surface to resolve the boundary layer, where the first grid point to the wall is located at a distance of 10^{-6}. To achieve a Mach number of $M_\infty = 0.15$ and a Reynolds number of $Re = 10^6$ based on the length 1 (the airfoil chord), the flow conditions defined on Tab. 4.2 are used. The turbulence models used are both the

Variable	Value
ρ_∞	1.4
$\boldsymbol{V}_\infty$	1
p_∞	44.44
μ_∞	2.33e-7

Spalart-Allmaras and k-ω SST models.

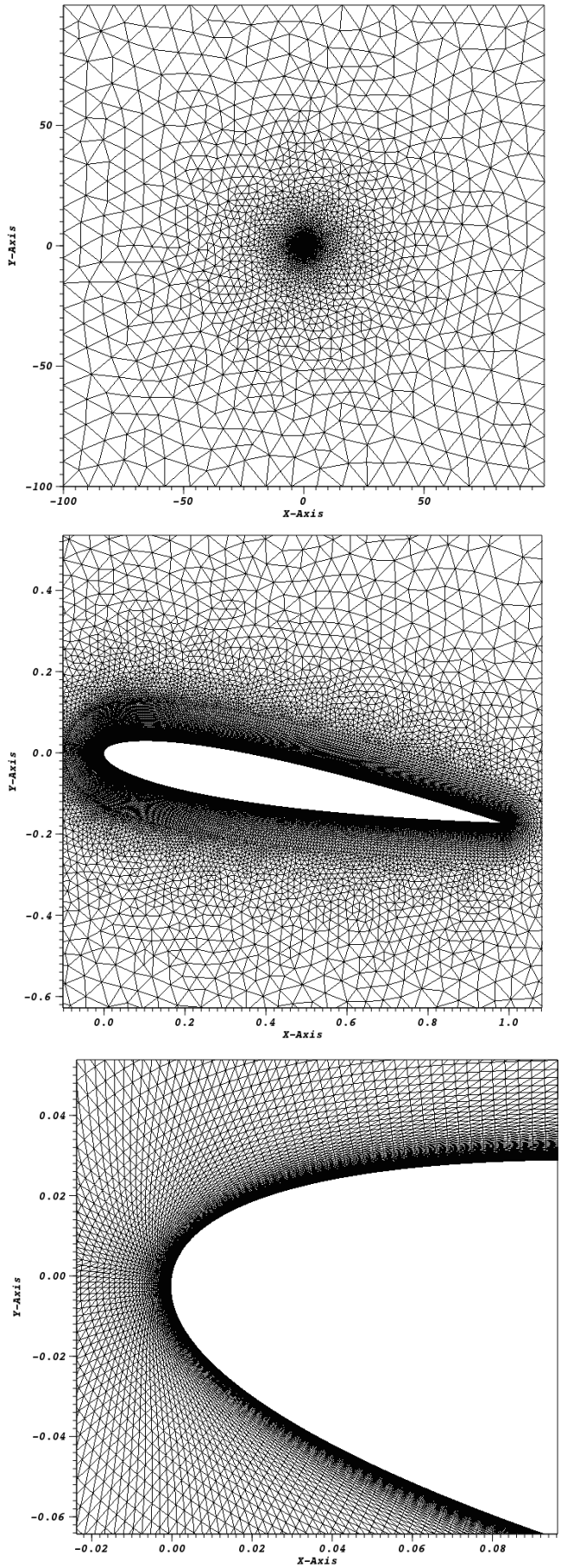


Figure 4.4: NACA0012 test case grid.

The residuals convergence history of the 15 degrees case for the Spalart-Allmaras and k-ω SST models are shown on Fig. 4.5 and Fig. 4.6.

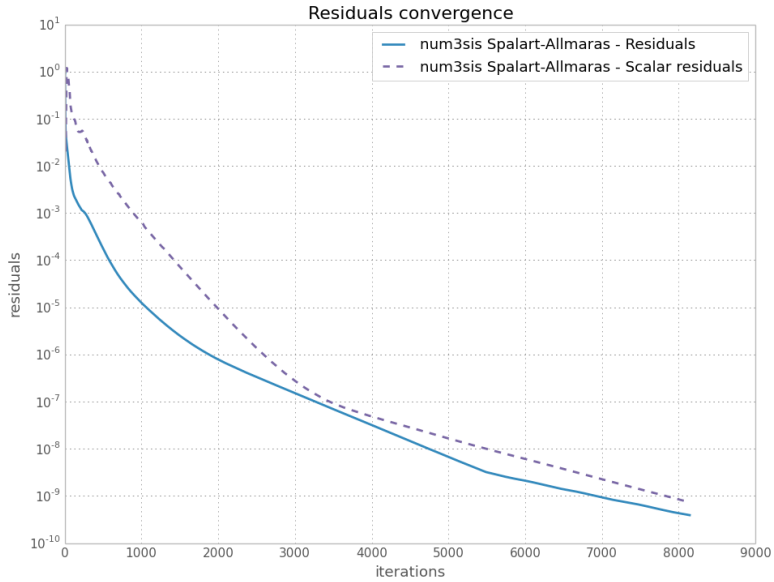


Figure 4.5: NACA0012 at 15 degrees residuals convergence using the Spalart-Allmaras turbulence model.

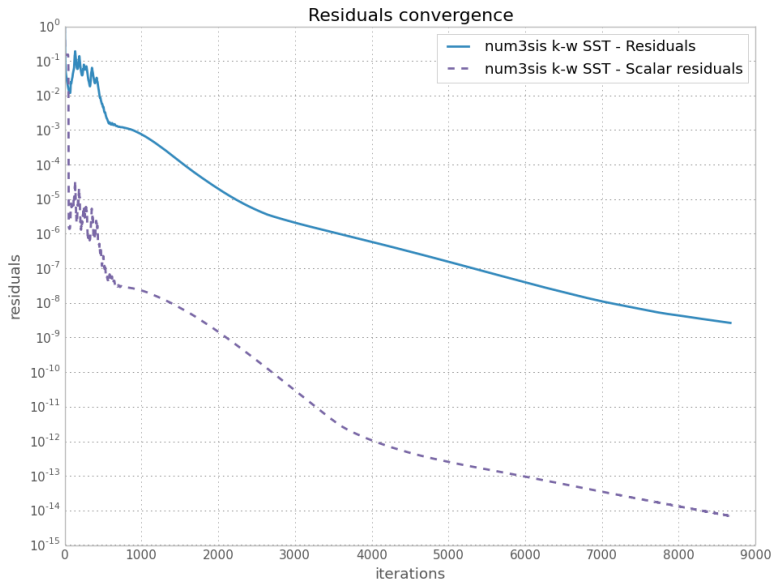
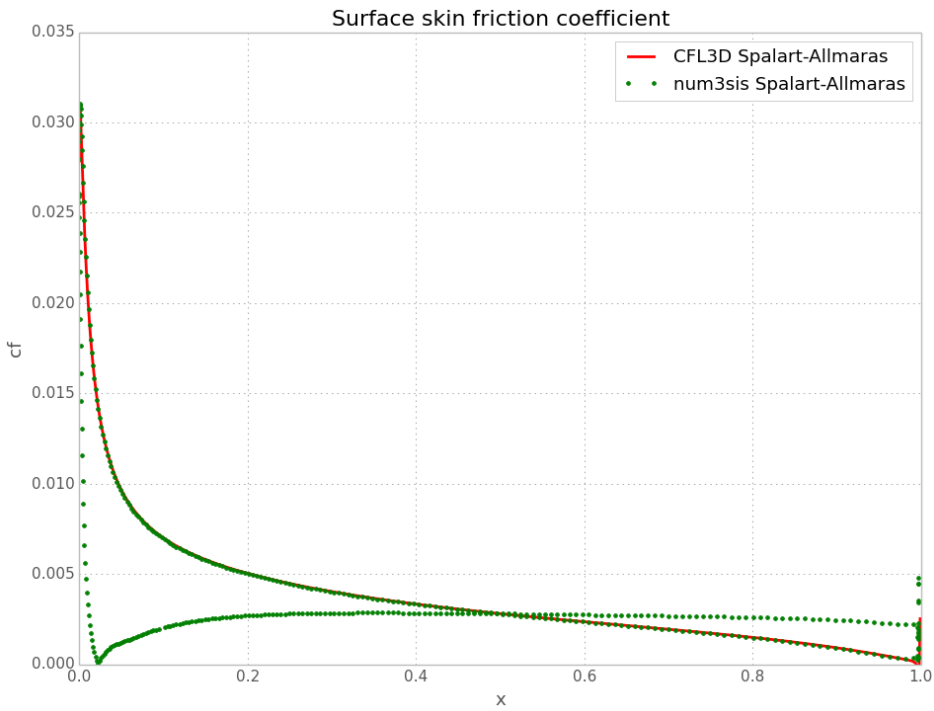


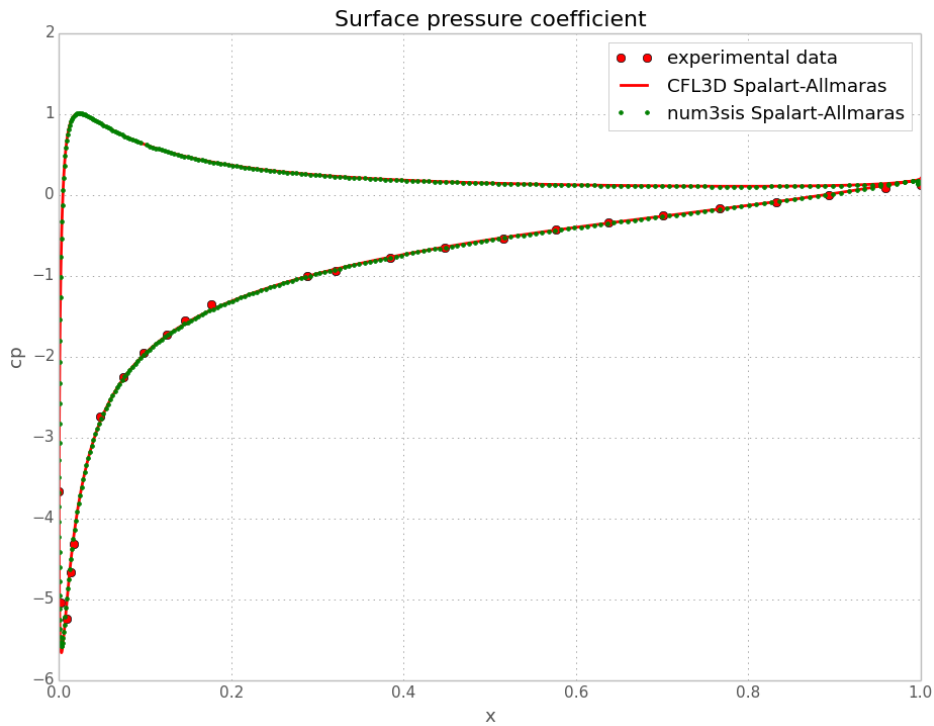
Figure 4.6: NACA0012 at 15 degrees residuals convergence using the k-ω SST turbulence model.

The friction and pressure coefficient distributions for the Spalart-Allmaras and k-ω SST models are presented on Fig 4.7, Fig. 4.8 for the 10 degrees case and Fig. 4.9, Fig. 4.10 for

the 15 degrees case. A small flow separation on the upper surface occurs at approximately 92% of the chord with the Spalart-Allmaras model and no separation with the k-ω SST model. The results show a good agreement with the CFL3D code on a 897×257 grid and the experimental data, validating the correct implementation of the turbulence models for subsonic flows around airfoils, when the flow is completely attached.

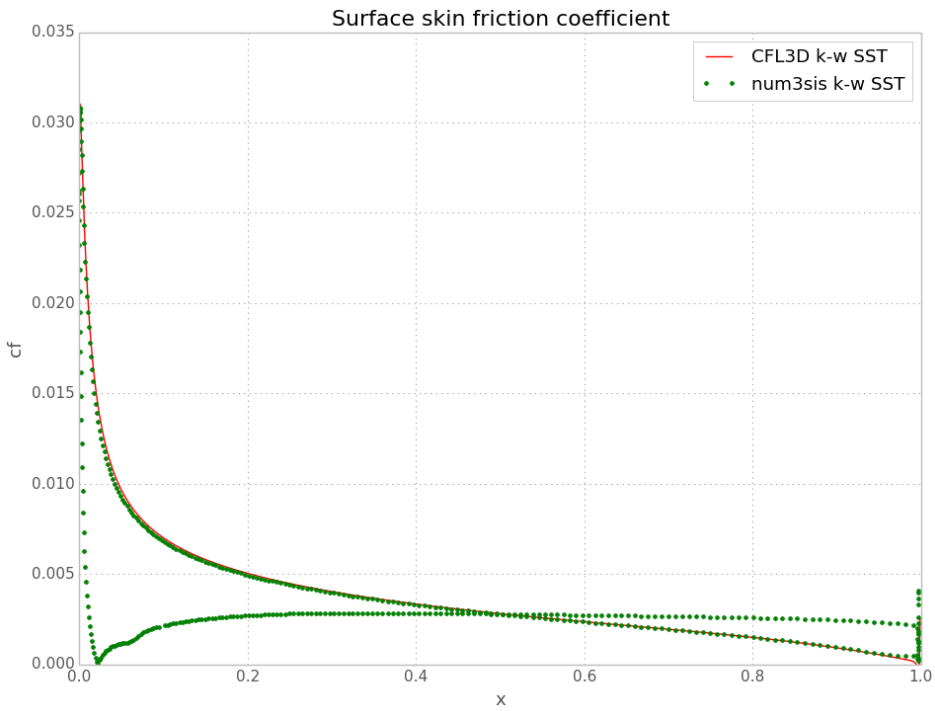


(a) Friction coefficient.

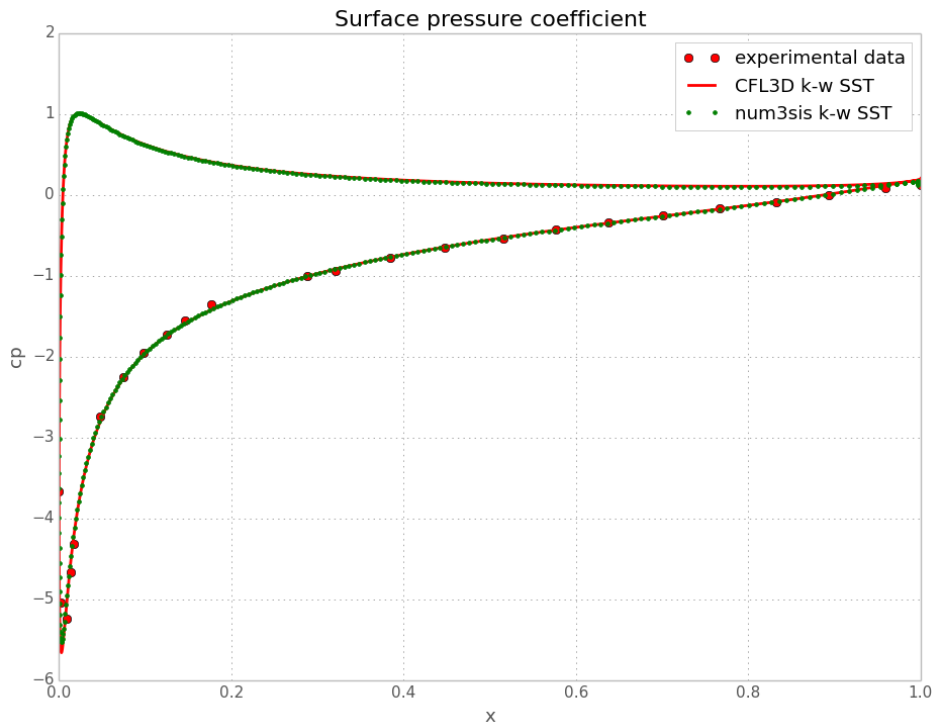


(b) Pressure coefficient.

Figure 4.7: NACA0012 at 10 degrees friction coefficient and pressure coefficient using the Spalart-Allmaras turbulence model.

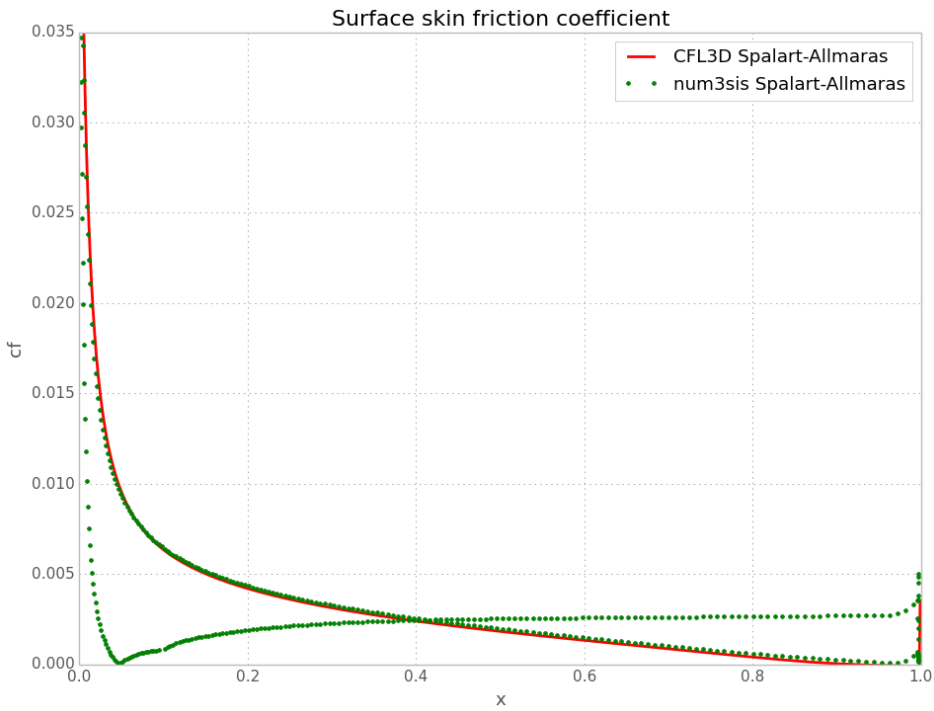


(a) Friction coefficient.

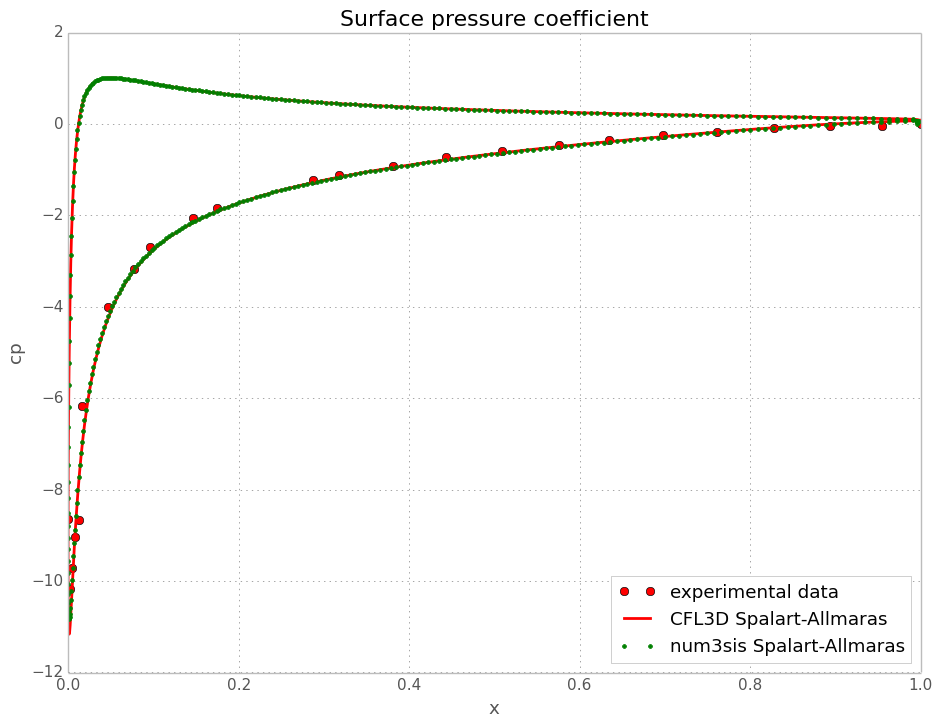


(b) Pressure coefficient.

Figure 4.8: NACA0012 at 10 degrees friction coefficient and pressure coefficient using the k-ω SST turbulence model.

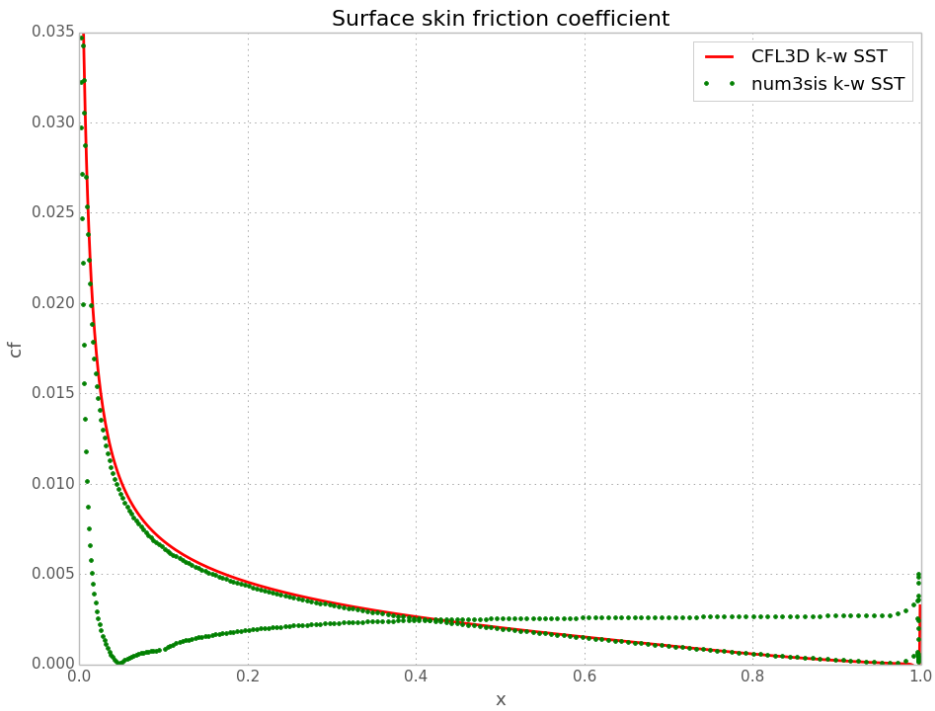


(a) Friction coefficient.

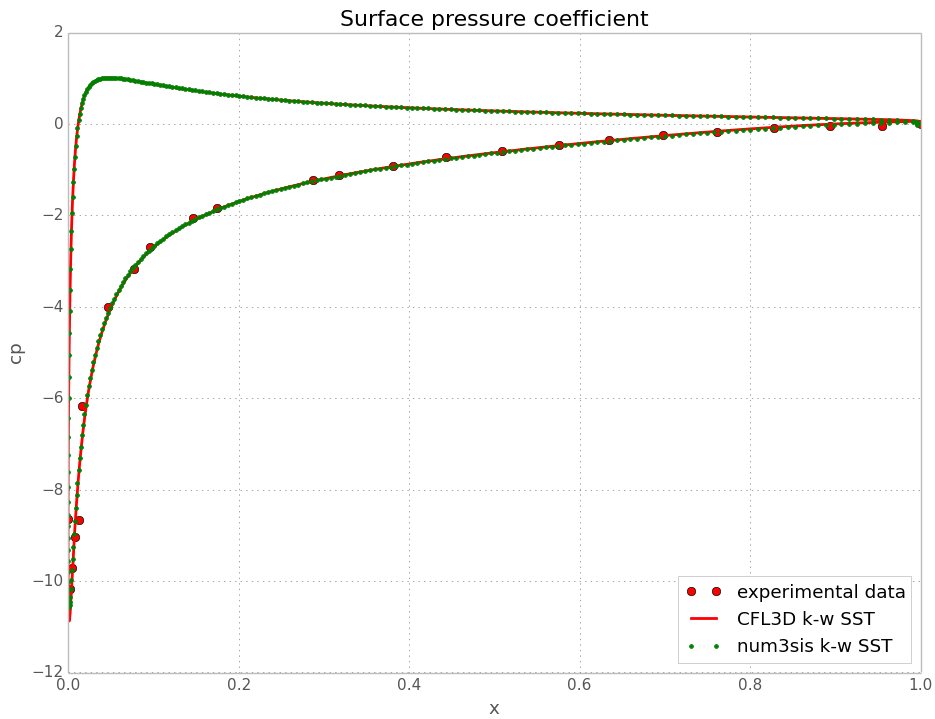


(b) Pressure coefficient.

Figure 4.9: NACA0012 at 15 degrees friction coefficient and pressure coefficient using the Spalart-Allmaras turbulence model.



(a) Friction coefficient.



(b) Pressure coefficient.

Figure 4.10: NACA0012 at 15 degrees friction coefficient and pressure coefficient using the k-ω SST turbulence model.

Chapter 5

Flow with actuation study

Contents

5.1 Comparison of the synthetic jet models 61
5.2 Impact of numerical parameters 80
5.3 Refinement study 81
5.4 Impact of turbulence closure 81
5.5 Synthesis 87

Steady non-controlled flow have been validated on reference test cases. This chapter focuses on a first study of unsteady actuated flows using different synthetic jet models. The prediction of the flow by the different models and the impact of the numerical parameters are quantified in this study.

To do so, a single synthetic jet located on the previously described flat plate and interacting a boundary layer is considered as a test-case, as illustrated in Fig. (5.1). The baseline case, without actuator, corresponds to the zero pressure gradient flat plate verification case proposed by NASA and presented in the previous chapter. The plate length is 2 m, and the computational domain height is 30 cm. The distance between the inlet boundary and the plate is 40 cm, and the distance between the stagnation point and the jet center is 50.25 cm. The slot width measures $h = 5$ mm. The length of the slot is twice its width $2h$. The cavity dimension is $9h \times h$.

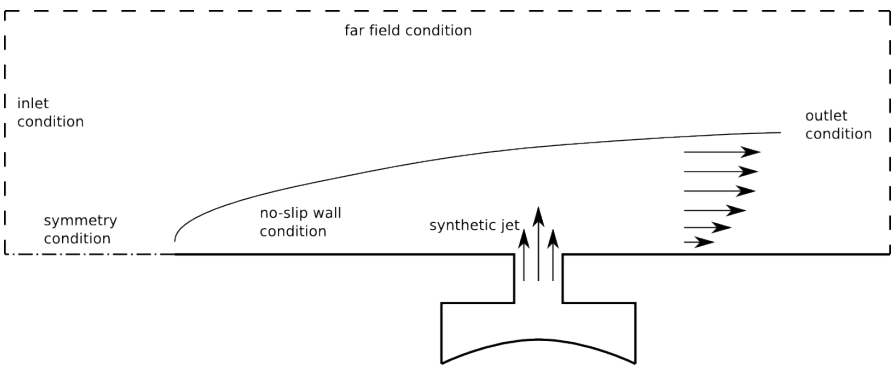


Figure 5.1: Flat plate with jet test-case description.

The reference flow conditions are the following:

ρ_{ref}	1.363 kg/m^3
u_{ref}	69.437 m/s
p_{ref}	115056 Pa
μ_{ref}	$1.9\,10^{-5}$ Pa s

The resulting Mach and Reynolds number are respectively $M_{ref} = 0.2$ and $Re_{ref} = 5\,10^6$. Note that the resulting boundary layer thickness at the actuation location is about twice the slot width.

All inlet/outlet boundary conditions are imposed weakly. At the inlet boundary, the velocity value u_{ref} and density ρ_{ref} are imposed, while the pressure is computed from the interior domain. On the contrary, an imposed pressure condition of value p_{ref} is prescribed at the outlet boundary. For the far-field condition, boundary values are computed thanks to Riemann invariants.

Two sets of actuation parameters are tested. The first one corresponds to a rather low frequency - low amplitude actuation, whereas the second one exhibits high frequency - high amplitude characteristics:

Actuation 1	$f = 50$ Hz	$Us = u_{ref}/2 = 34.72$ m/s
Actuation 2	$f = 500$ Hz	$Us = 2\,u_{ref} = 138.87$ m/s

For all the computations below, the time-step is chosen in order to account for 200 steps for each actuation period. Therefore, the time-step is defined as $\Delta t_1 = 1.\,10^{-4}$ s for the first actuation, and $\Delta t_2 = 1.\,10^{-5}$ s for the second one. The unsteady simulations are initialized by the steady state solutions corresponding to the flows without actuation. For each time-step, a stopping criterion corresponding to a reduction of 3 orders of the non-linear residuals is adopted.

The baseline grids used for the three models are depicted on Figs. (5.2-5.5) and count $N_c = 36848$, $N_s = 19304$ and $N_b = 17424$ vertices for the boundary model, the slot model and the cavity model respectively. The maximal aspect ratio of the cells is about 40000. The number of vertices located on the jet exit is 40. Note that the grids are identical in the outer domain.

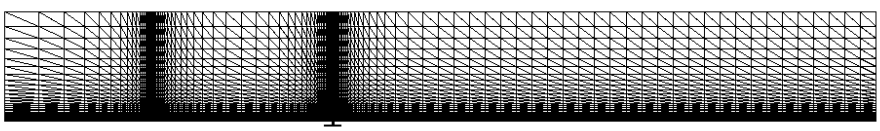


Figure 5.2: Global view of the mesh.

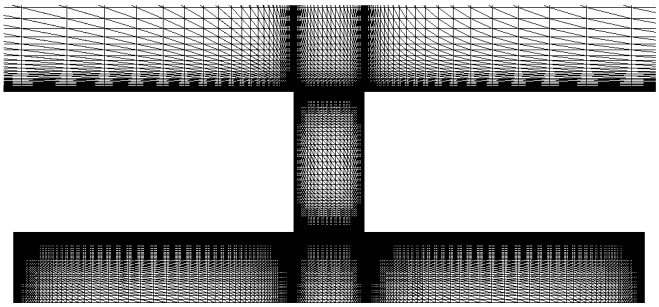


Figure 5.3: Mesh in the vicinity of the actuator, cavity model.

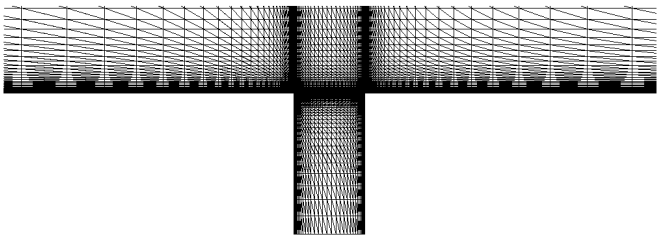


Figure 5.4: Mesh in the vicinity of the actuator, slot model.

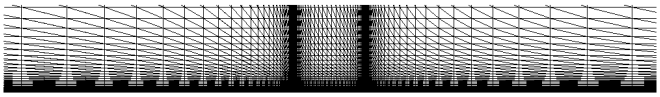


Figure 5.5: Mesh in the vicinity of the actuator, boundary model.

5.1 Comparison of the synthetic jet models

Computations are carried out until a periodic flow is observed. Comparisons of the two velocity components for the three models, in the vicinity of the actuator, are provided by Figs. (5.6-5.13) for the first actuation, and Figs. (5.14-5.21) for the second actuation. On these figures, the phase $\Phi = 0$ corresponds to the maximum blowing time and $\Phi = \pi$ to the maximum suction time.

Clearly, the flows obtained using the three models are close to each other. The boundary model generates obviously a more symmetric flow at the slot exit, due to the boundary condition on the velocity. The slot and the cavity models allow to compute the flow in the slot, which is characterized by strong asymmetry and generates a more intense flow at the slot corners. One can notice that the cavity and slot models only differ at the bottom part of the slot, with a negligible influence on the flow in the outer domain. The second actuation, with high frequency and amplitude, exhibits larger discrepancies.

A comparison of the drag coefficient is also performed, as a more global assessment criterion. Fig. (5.22) confirms that the cavity and boundary models predict very similar

flows, whereas the boundary model slightly underestimates the drag coefficient value, especially for the second actuation parameters.

The conclusions of these comparisons are the following: although the cavity model is far more CPU-demanding than the slot model, the discrepancy between the two predicted flows is weak, in terms of local field values and global drag coefficient. The boundary model predicts similar flows. However, some differences are reported for high-frequency high-amplitude actuation. In the perspective of more complex studies, the slot model seems to be the best compromise, in terms of CPU cost and flow prediction. An alternate approach could be to capture the velocity profile computed at the slot exit using the cavity model, and use it as boundary condition for the boundary model. However, if one considers a jet with varying parameters (amplitude, frequency, location), it is not clear that the selected profile will correspond to the new conditions.

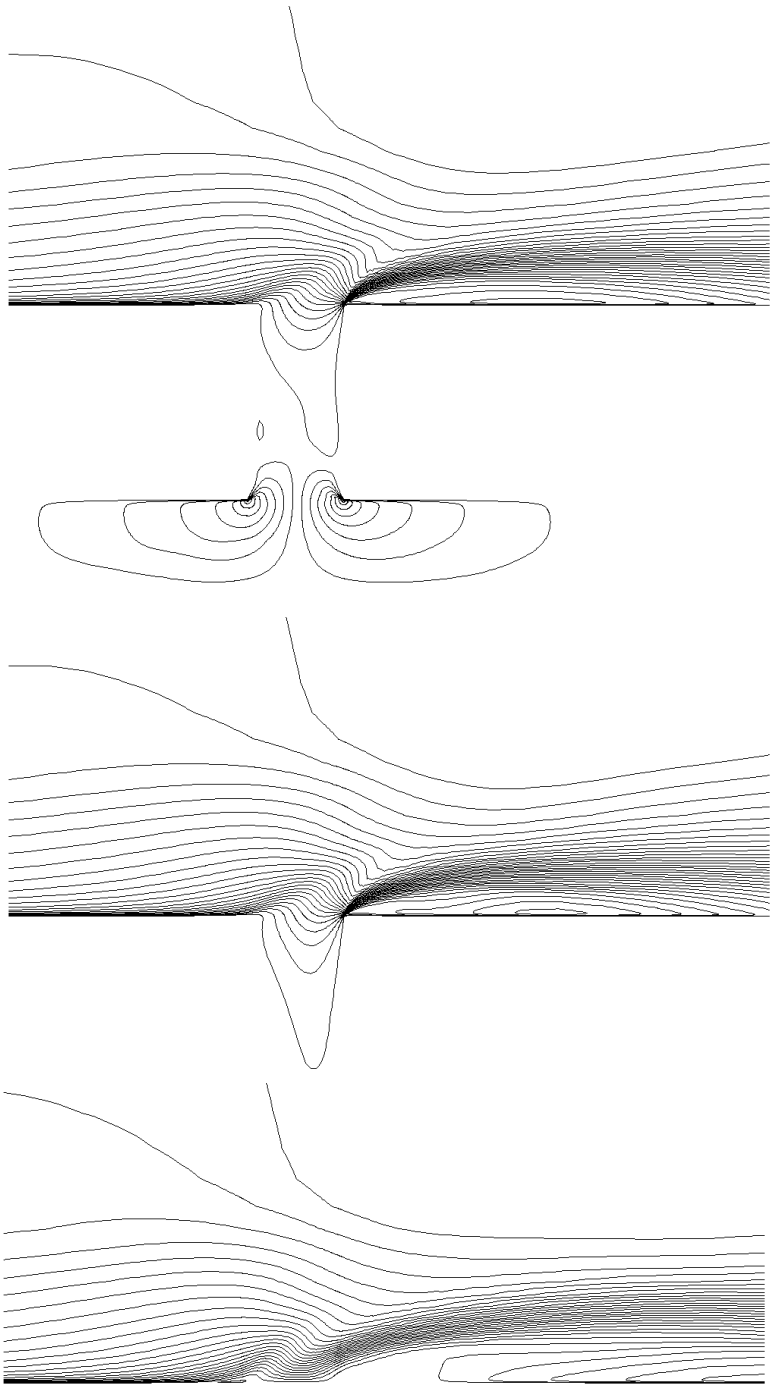


Figure 5.6: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 0$ (50 Hz).

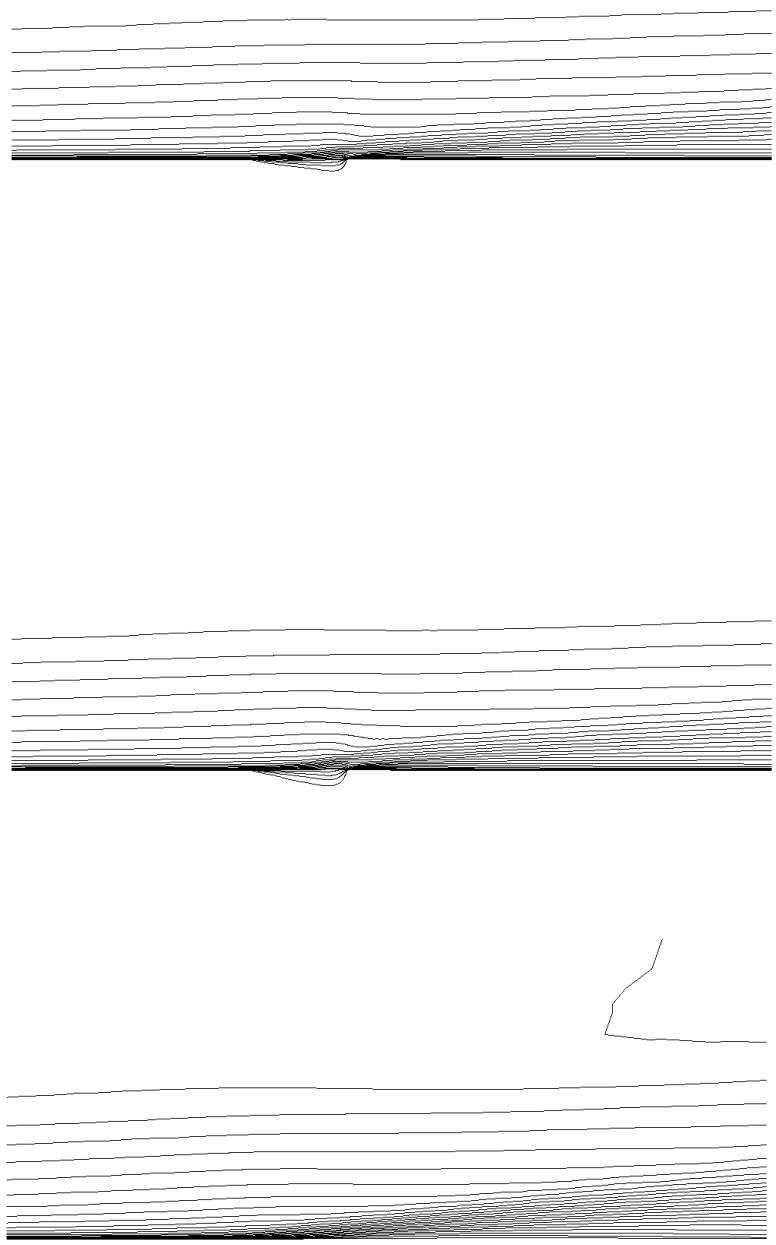


Figure 5.7: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi/2$ (50 Hz).

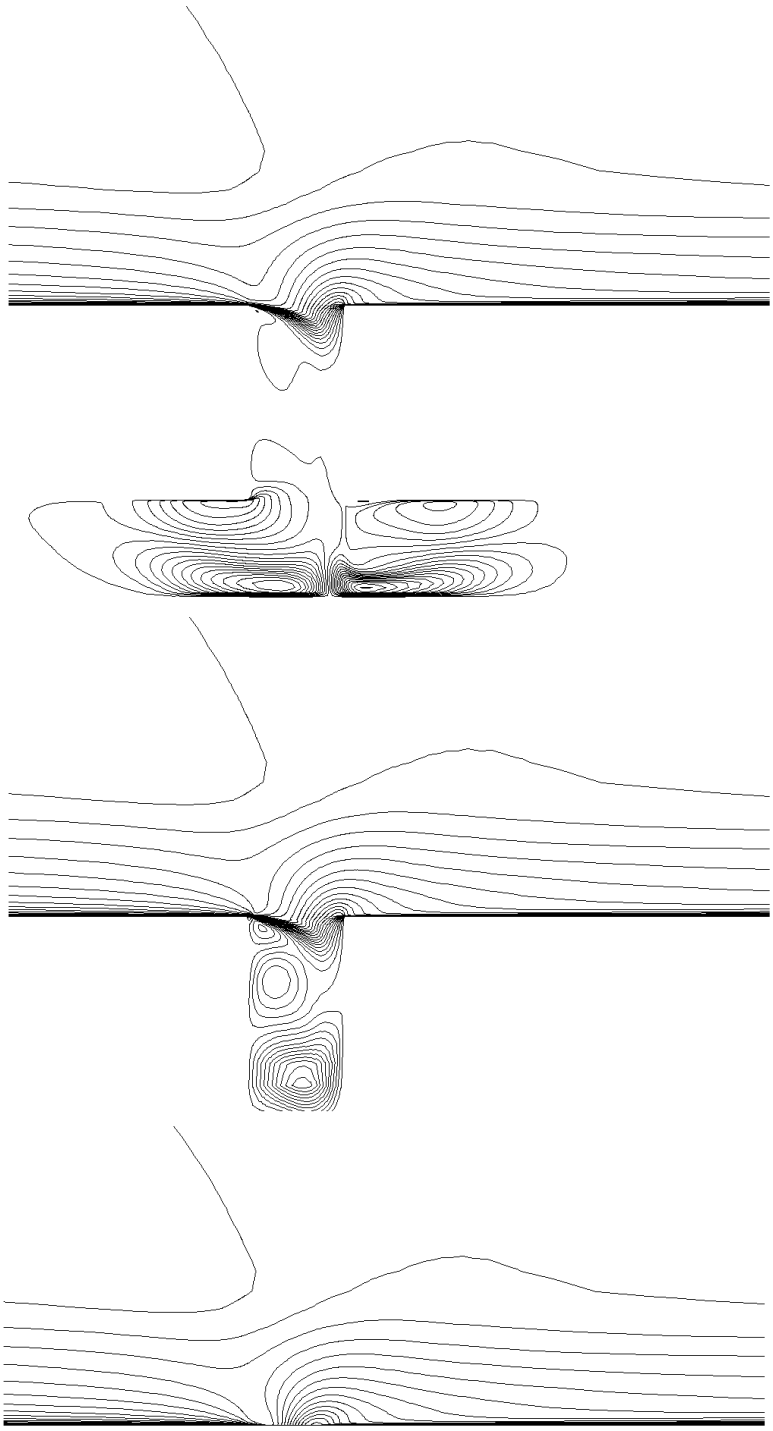


Figure 5.8: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi$ (50 Hz).

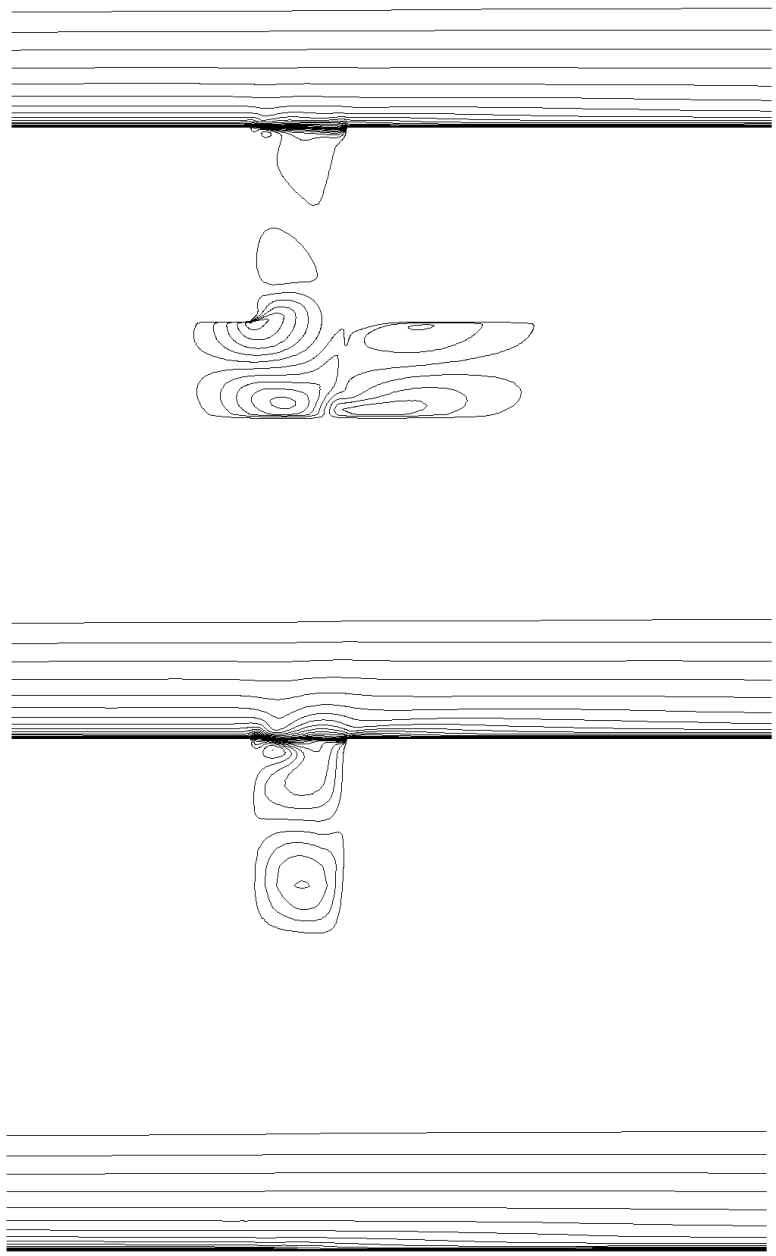


Figure 5.9: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 3\pi/2$ (50 Hz).

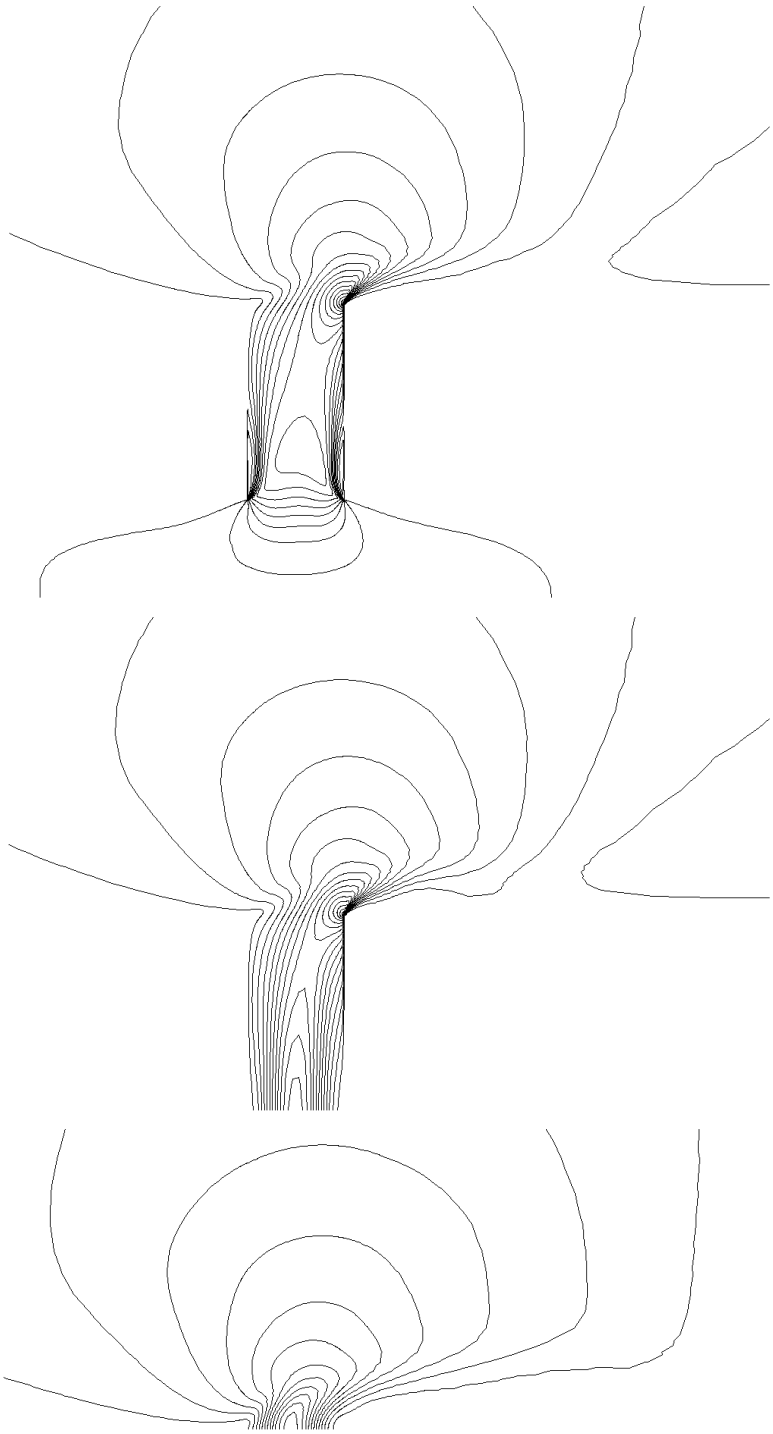


Figure 5.10: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 0$ (50 Hz).

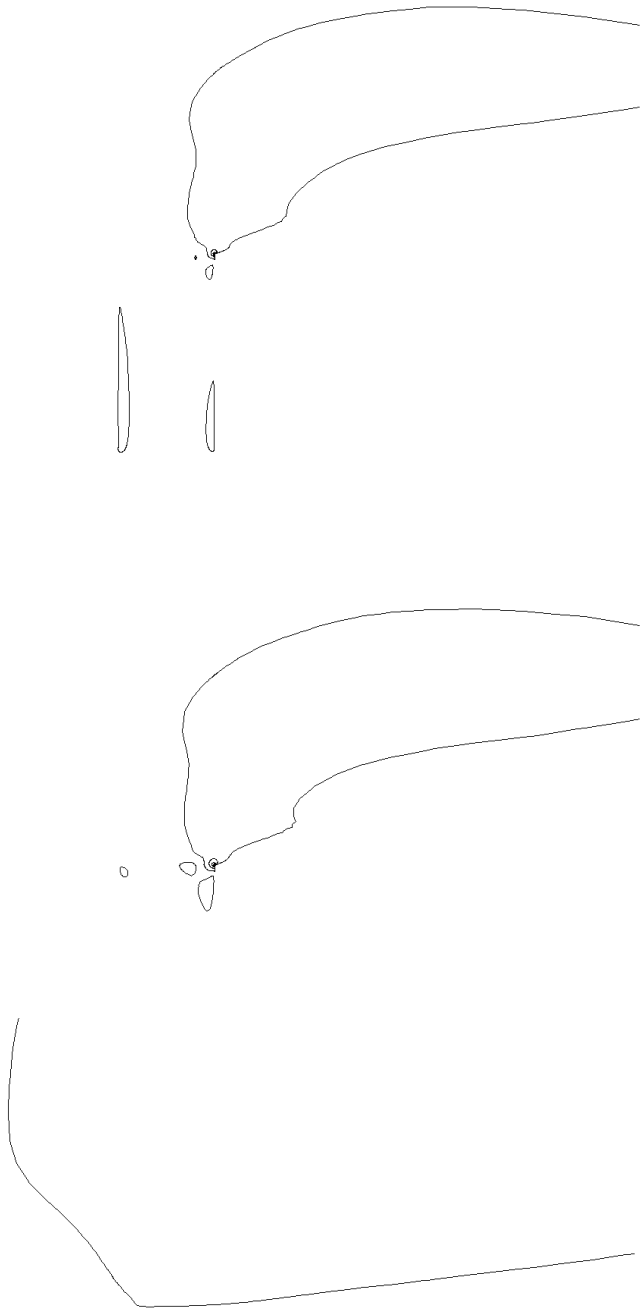


Figure 5.11: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi/2$ (50 Hz).

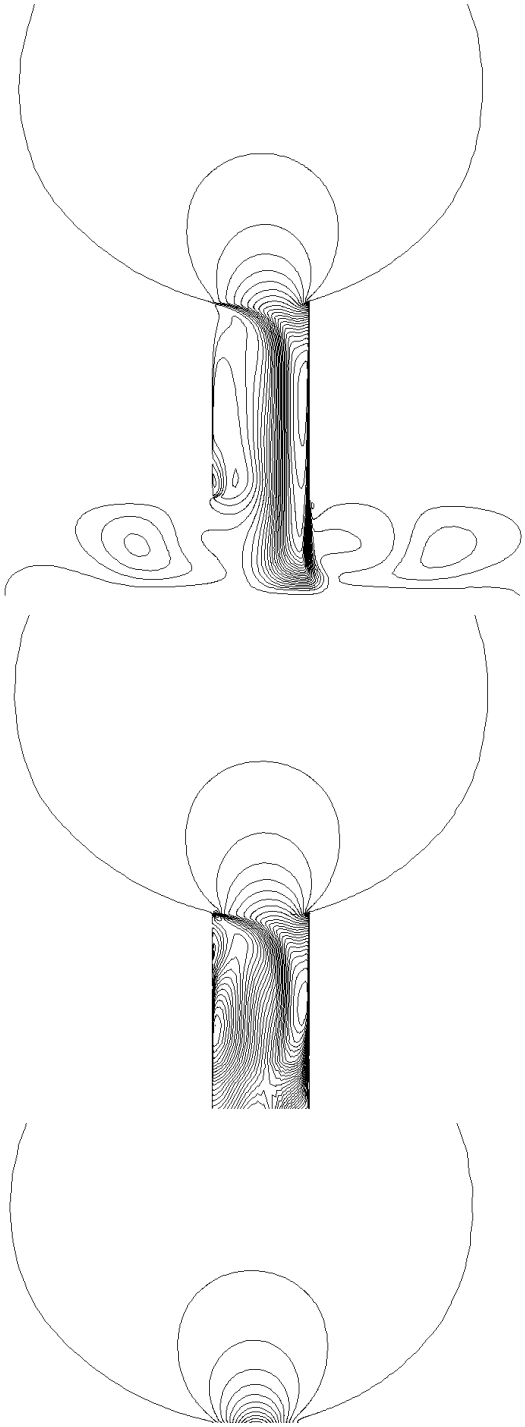


Figure 5.12: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi$ (50 Hz).

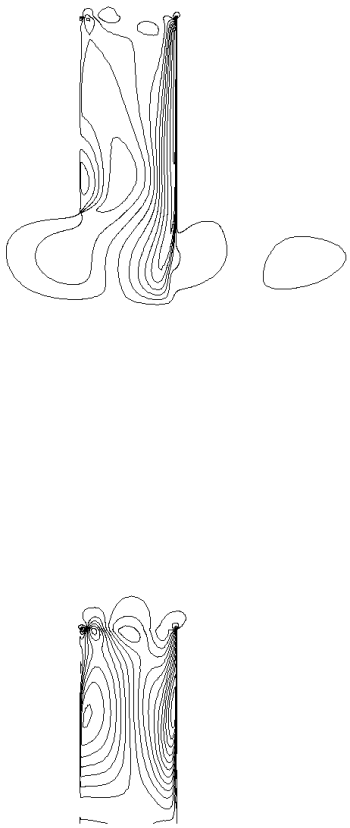


Figure 5.13: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 3\pi/2$ (50 Hz).

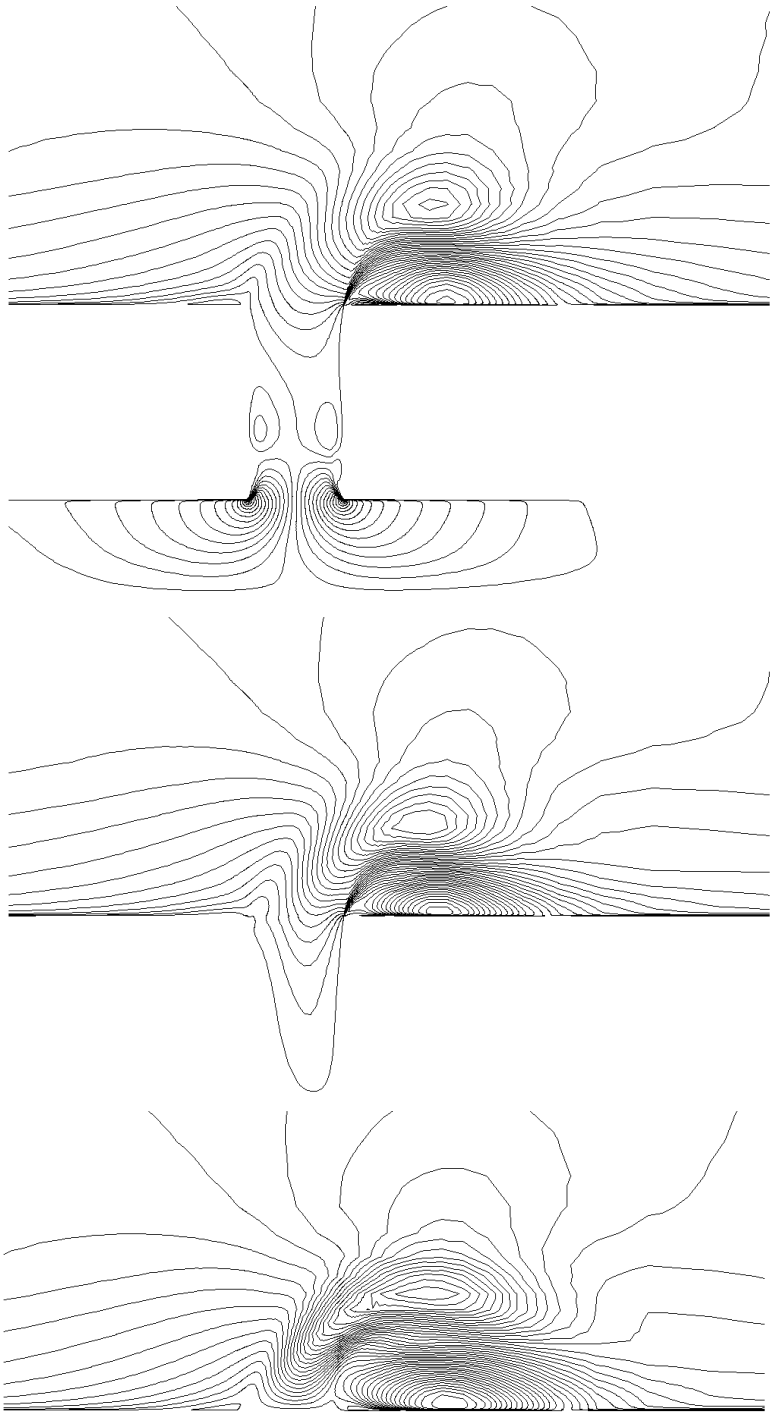


Figure 5.14: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 0$ (500 Hz).

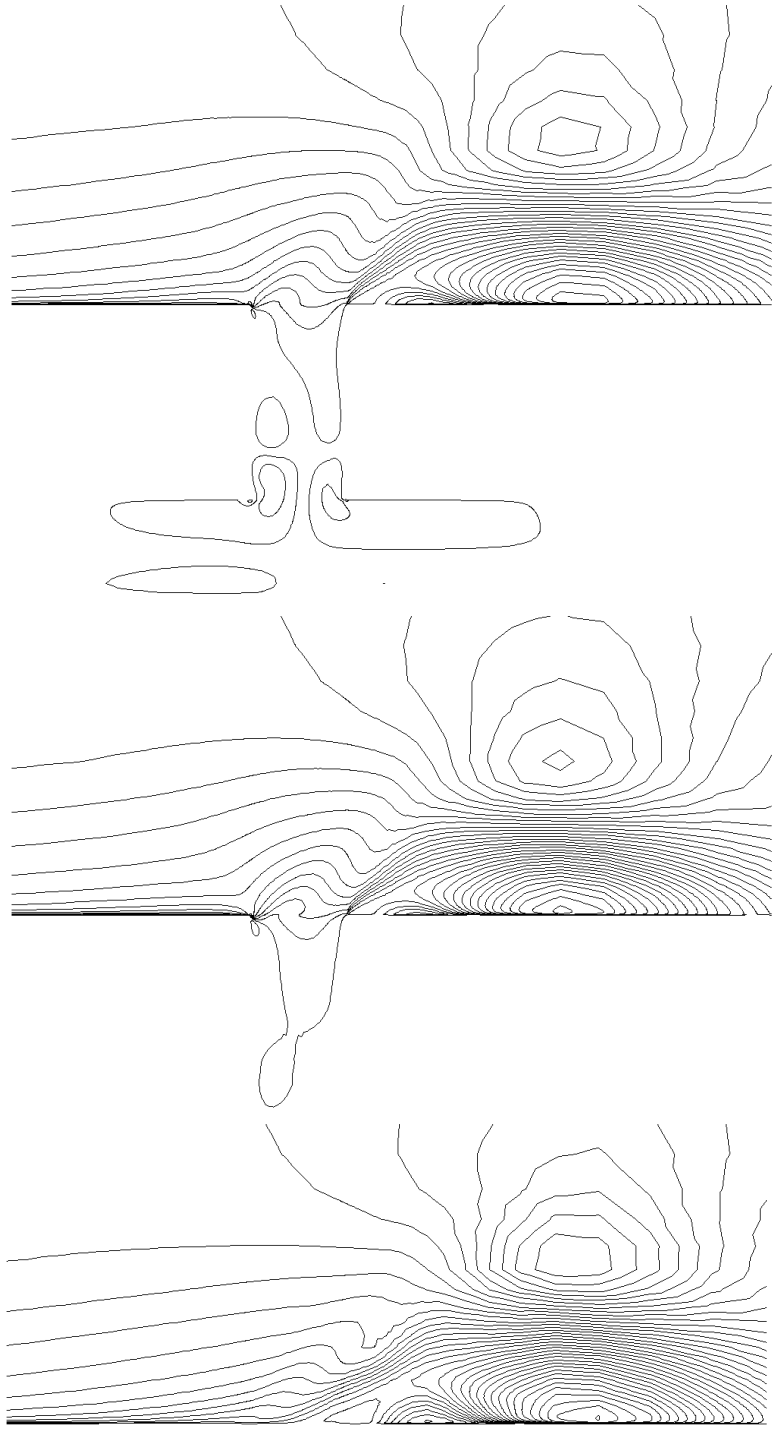


Figure 5.15: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi/2$ (500 Hz).

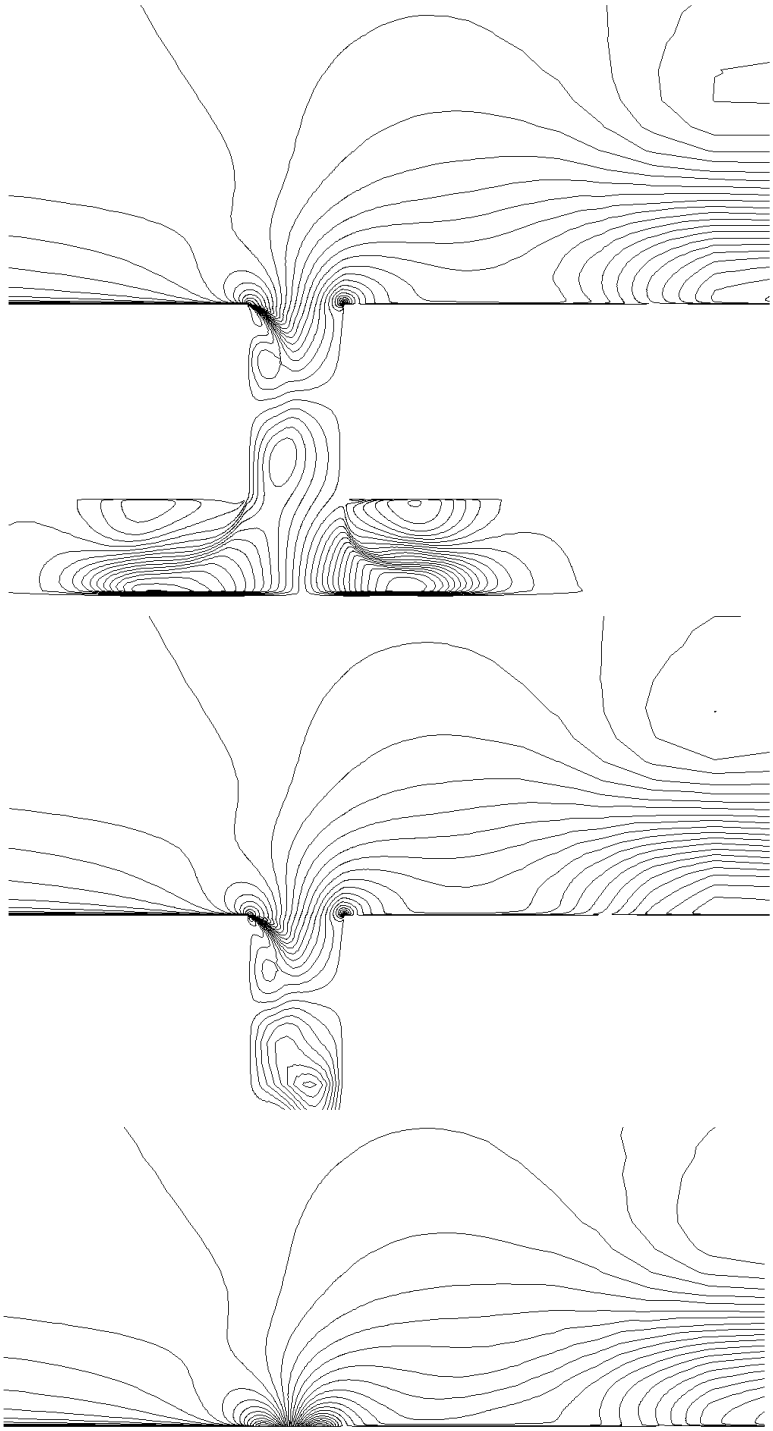


Figure 5.16: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi$ (500 Hz).

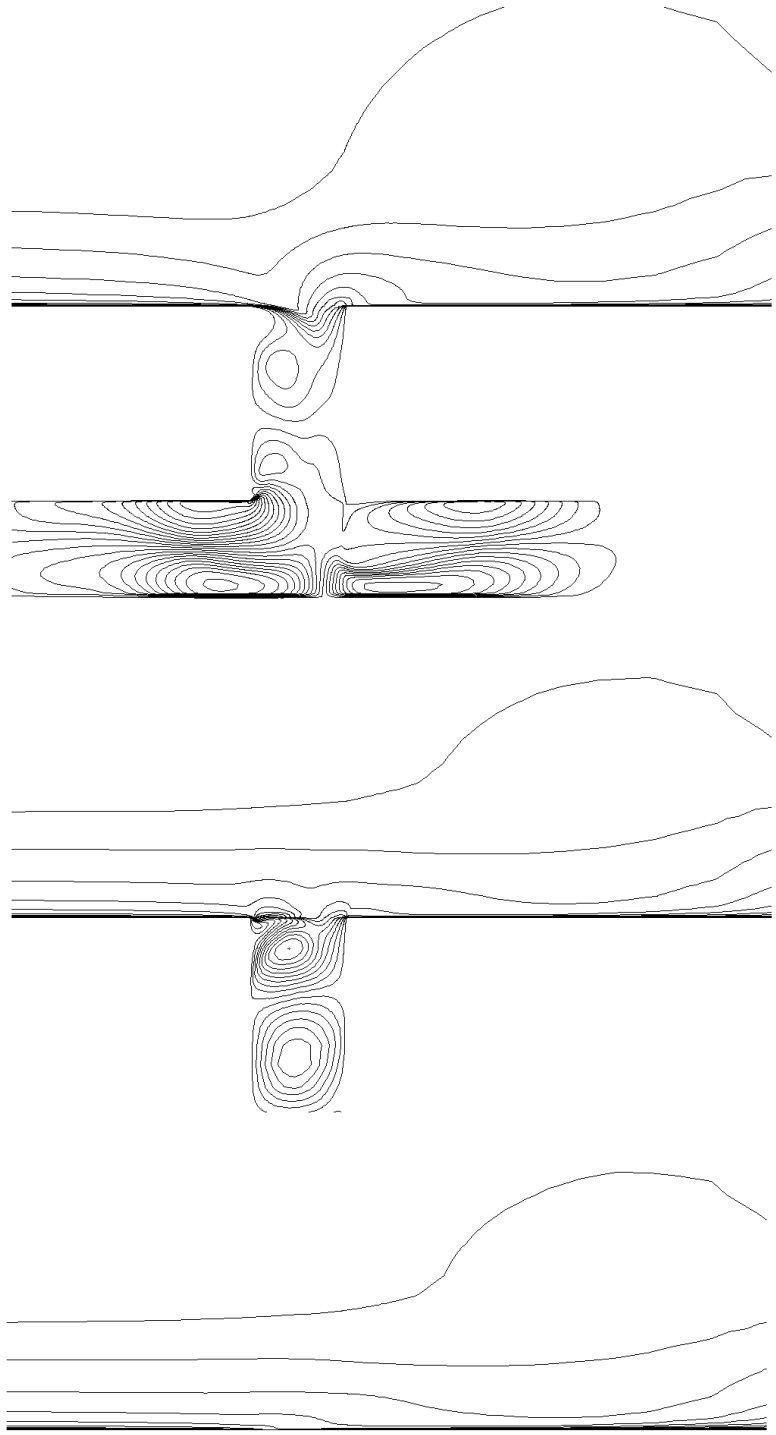


Figure 5.17: Iso-u contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 3\pi/2$ (500 Hz).

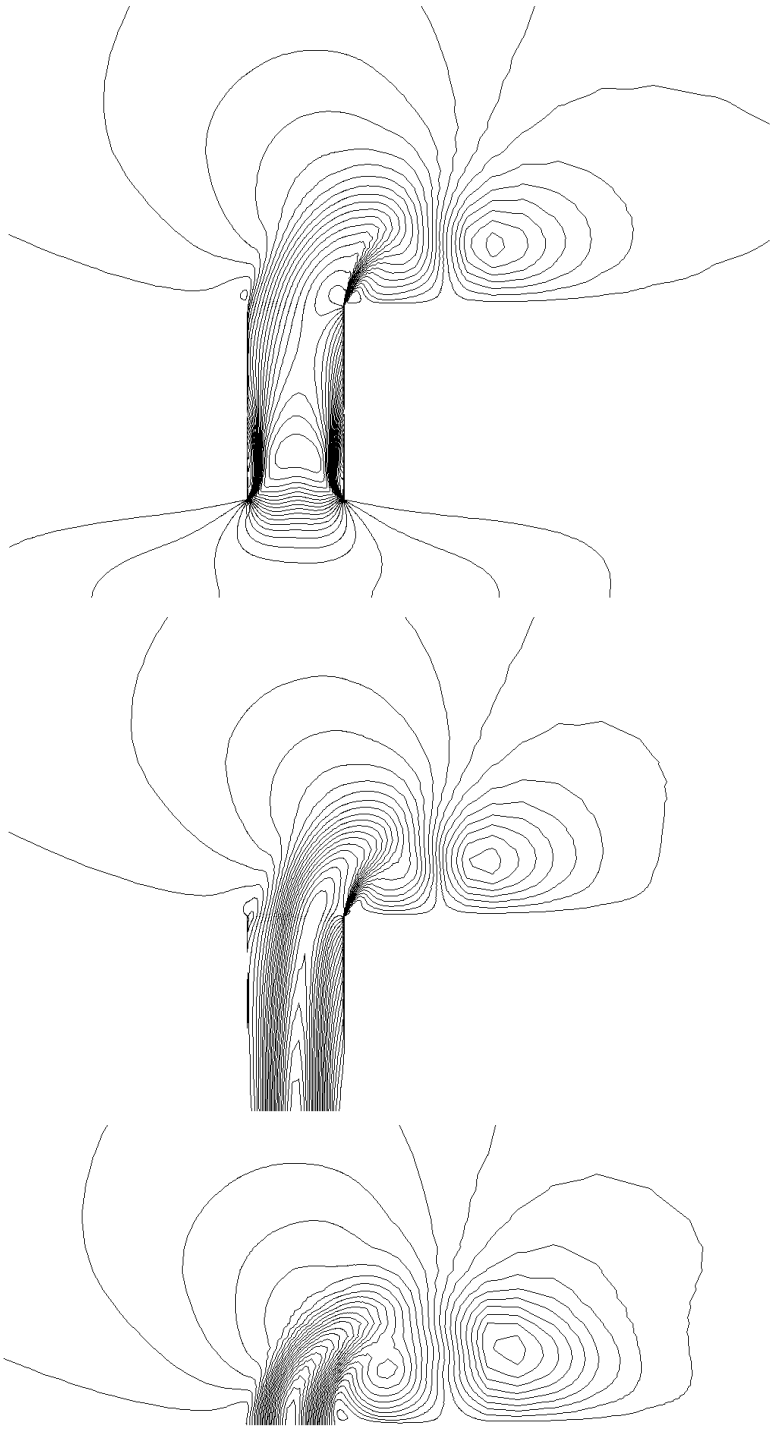


Figure 5.18: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 0$ (500 Hz).

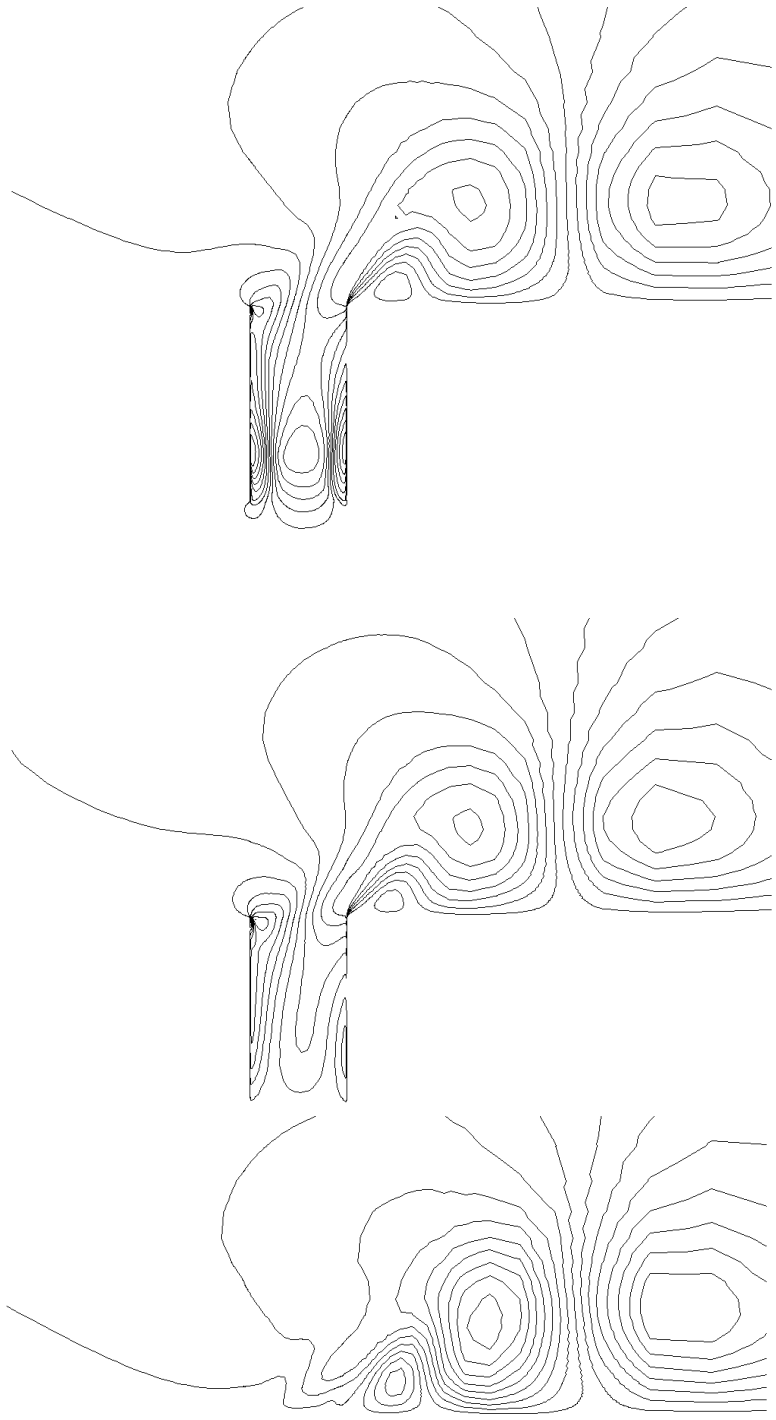


Figure 5.19: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi/2$ (500 Hz).

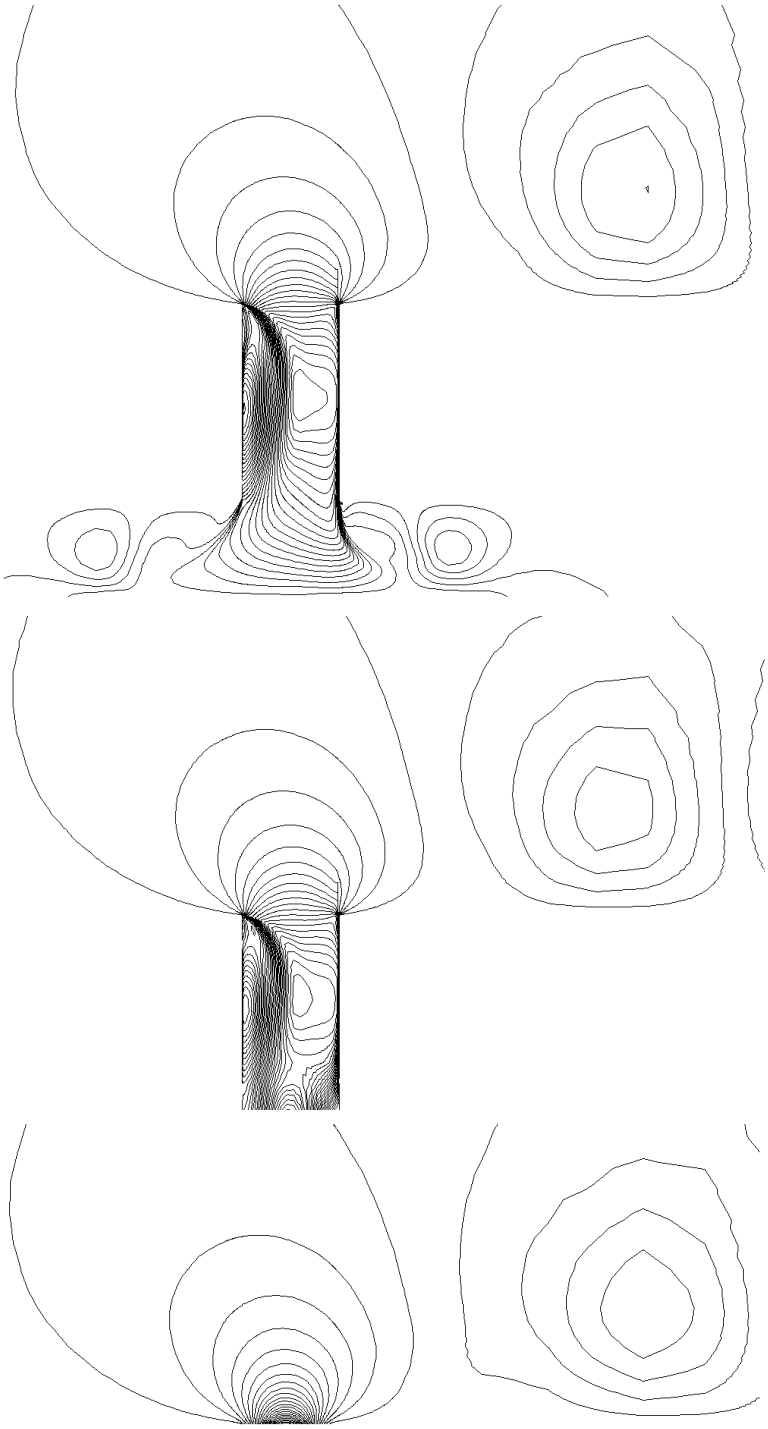


Figure 5.20: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = \pi$ (500 Hz).

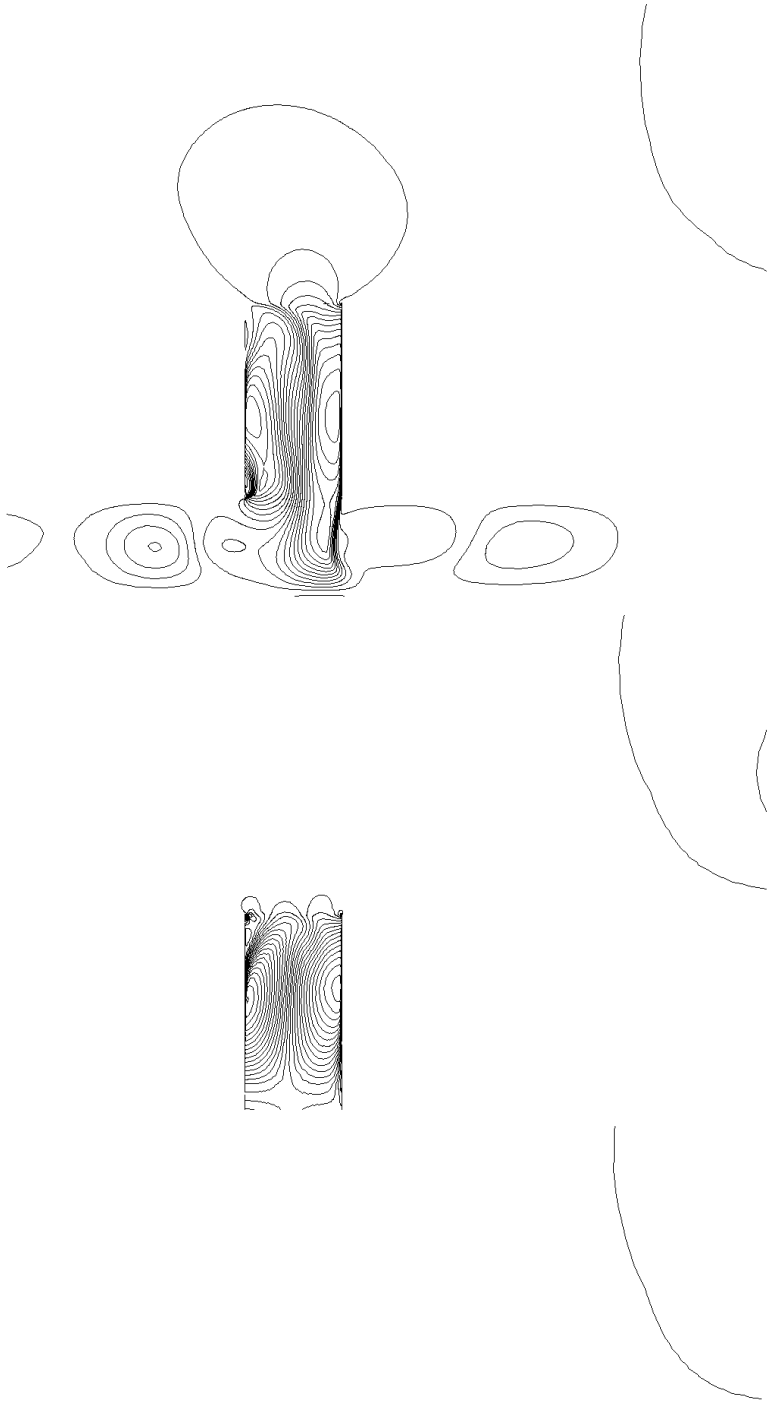


Figure 5.21: Iso-v contours for the cavity (top), slot (middle) and boundary (bottom) models, for $\Phi = 3\pi/2$ (500 Hz).

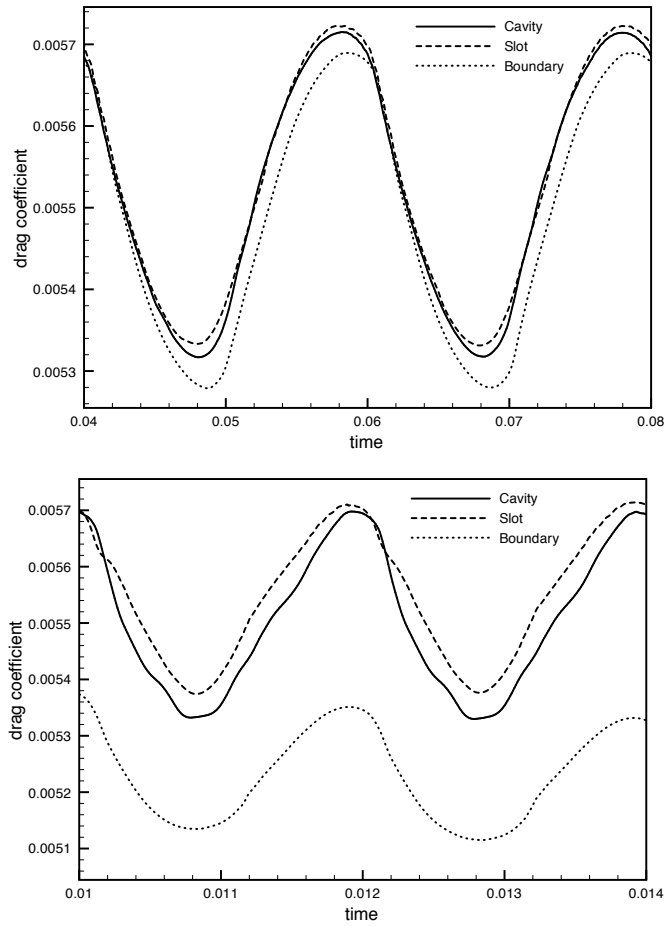


Figure 5.22: Time evolution of the drag coefficient, for the first actuation (top) and second actuation (bottom).

5.2 Impact of numerical parameters

In this section, the influence of some numerical parameters on the flow prediction is investigated. In the perspective of design optimization, this study is restricted to the boundary condition model for the actuation 1 and the drag coefficient is used as the main comparison criterion. More precisely, the influence of the choice of the boundary condition type for the jet (weak vs. strong) is tested and the impact of the convergence criterion used for each time step is measured.

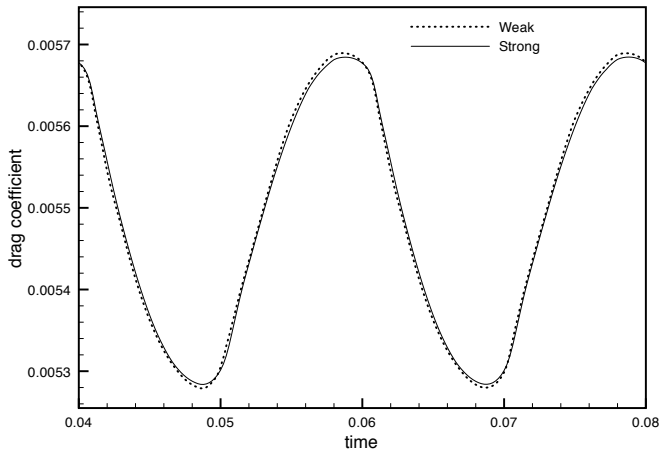


Figure 5.23: Time evolution of the drag coefficient, for weak and strong boundary conditions.

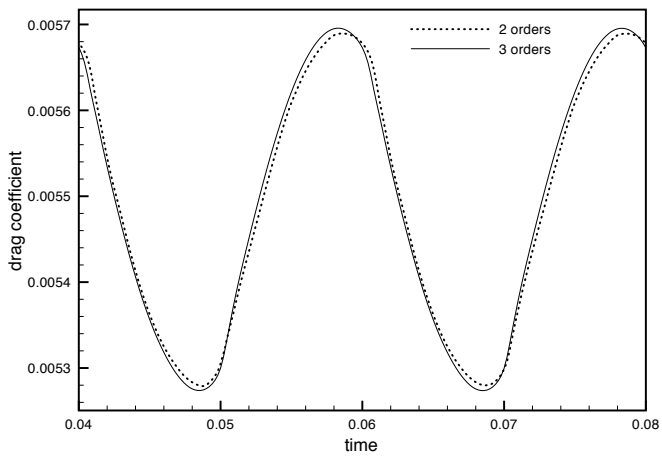


Figure 5.24: Time evolution of the drag coefficient, for different non-linear convergence criteria.

The evolution of the drag coefficient computed using weak and strong boundary conditions is depicted in Fig. (5.23). As seen, the discrepancy is not relevant. Indeed, the

iterative process carried out at each time-step makes the two approaches nearly identical.

Fig. (5.24) shows the same quantity, when different parameters are used as stopping criterion for the non-linear iterative process. Here again, the discrepancy is small, which indicates that the flow at each time-step is well converged.

5.3 Refinement study

To assess the computations, refinement study is finally performed, by reducing the physical time step by a factor two, while applying a uniform grid refinement by a factor $\sqrt{2}$ in each direction, yielding a fine mesh counting 32010 vertices. The comparison of the drag coefficient histories for the two meshes is depicted in Fig. (5.25). As can be observed, the use of finer grid and time-step slightly modifies the drag coefficient value during the suction phase. However, the change is rather moderate. In particular, the time-averaged drag coefficient is not significantly affected by the refinement.

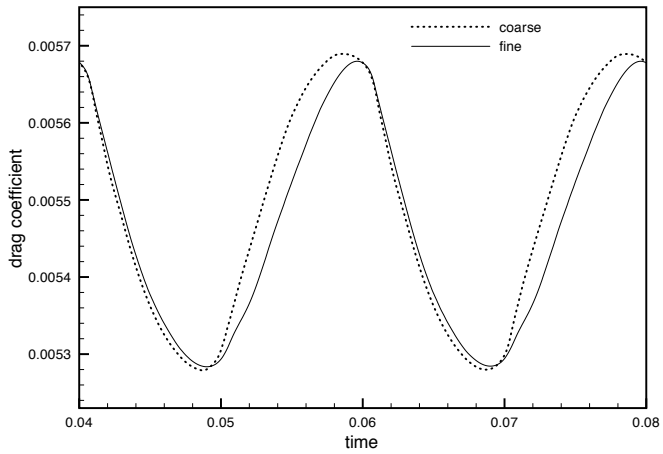


Figure 5.25: Time evolution of the drag coefficient, for two meshes.

5.4 Impact of turbulence closure

As the synthetic jet generates detached flows, it is of particular interest to compare the flows predicted by different turbulent closures. We consider for this study the SST k-ω from Menter and the Spalart-Allmaras closures, using the boundary model for both actuations.

The time evolution of the drag coefficient for the two models is represented in Fig (5.26). As seen, the SST k-ω closure predicts a lower drag than the Spalart-Allmaras one, especially during the blowing phase and for the high-frequency actuation. Nevertheless,

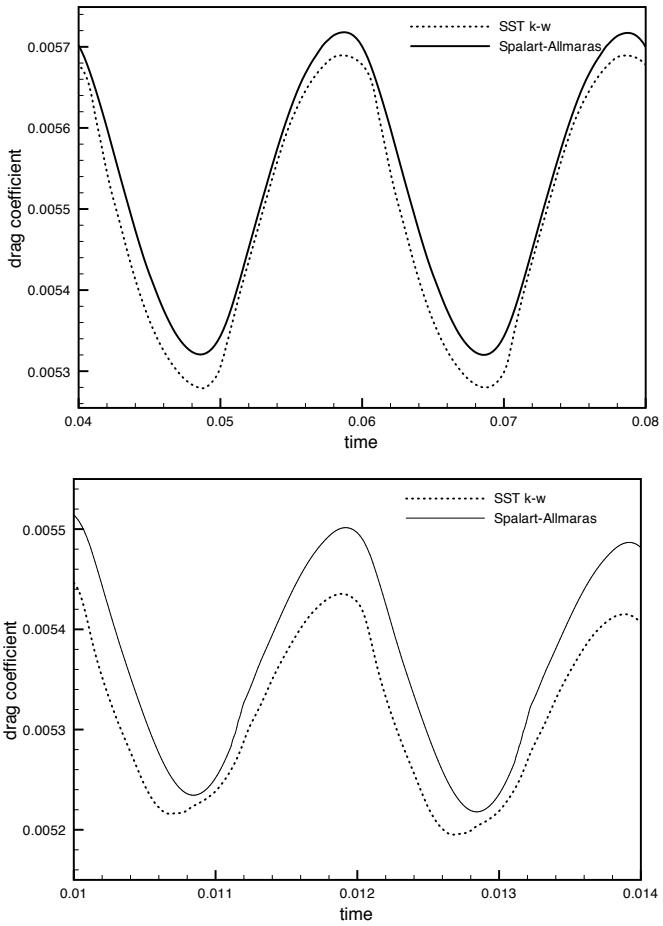


Figure 5.26: Time evolution of the drag coefficient for different turbulence closures, for the first actuation (top) and second actuation (bottom).

the discrepancy between these values is of the same magnitude than that observed for the steady flow without actuation (about 10^{-4}).

Figs (5.27-5.34) show velocity contours for the two closures at some actuation phases (maximum blowing and maximum suction). One can notice that the flows are globally quite close to each other. It is confirmed that they differ mainly during the blowing phase, the SST k-ω closure generating a more vortical flow, with more intense gradients, than the Spalart-Allmaras one.

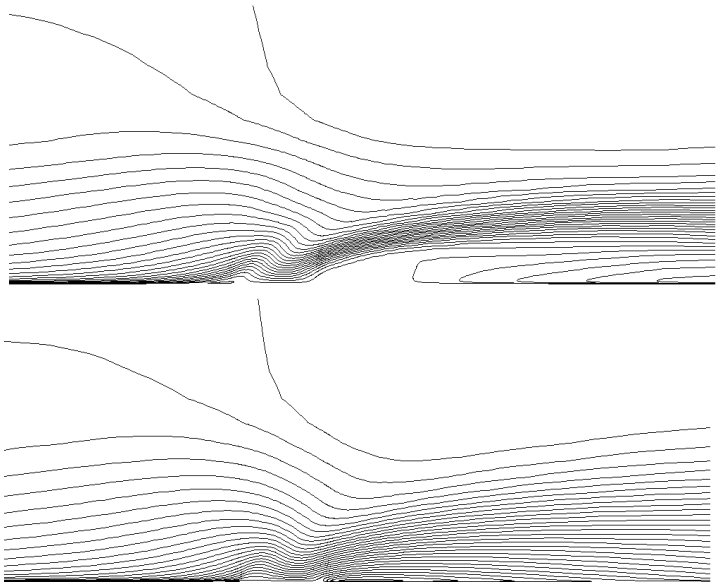


Figure 5.27: Iso-u contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 0$ (50 Hz).

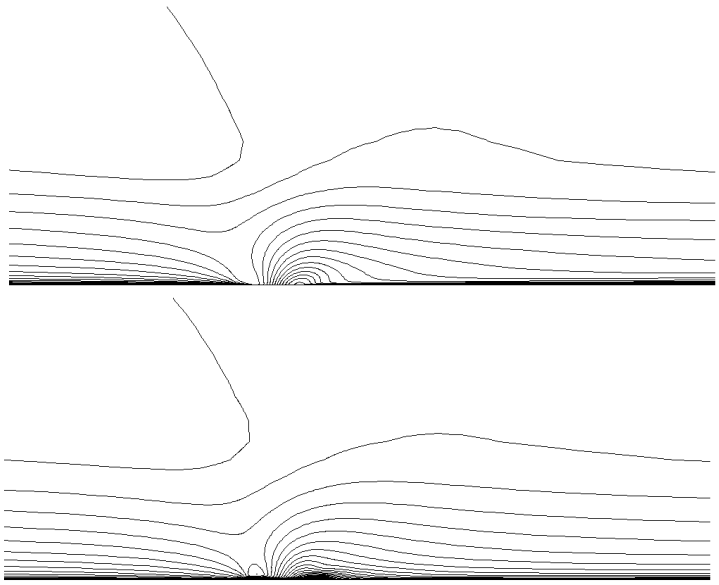


Figure 5.28: Iso-u contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 2$ (50 Hz).

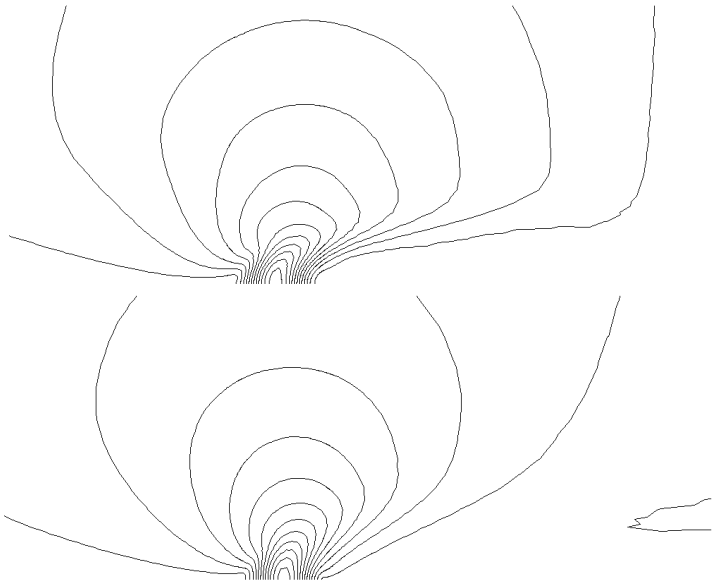


Figure 5.29: Iso-v contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 0$ (50 Hz).

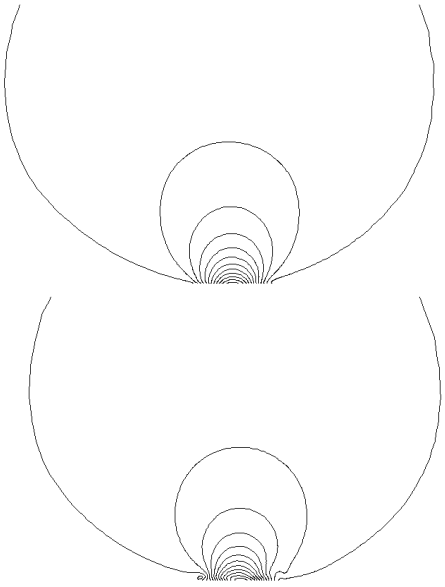


Figure 5.30: Iso-v contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 2$ (50 Hz).

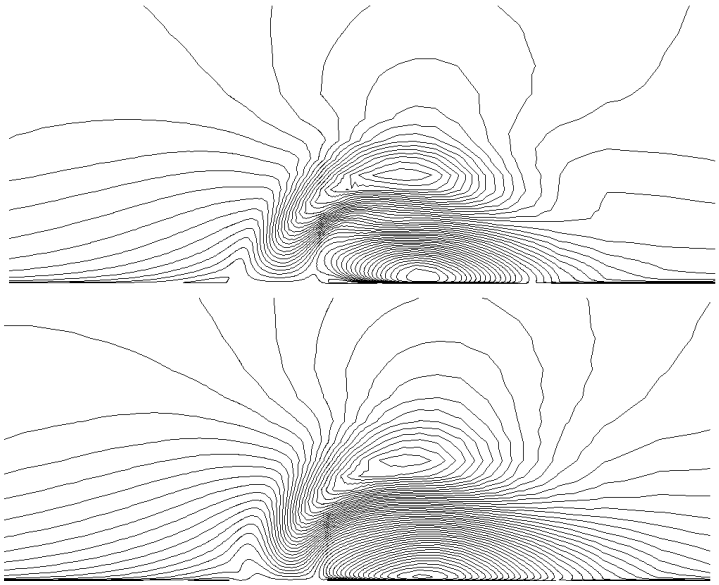


Figure 5.31: Iso-u contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 0$ (500 Hz).

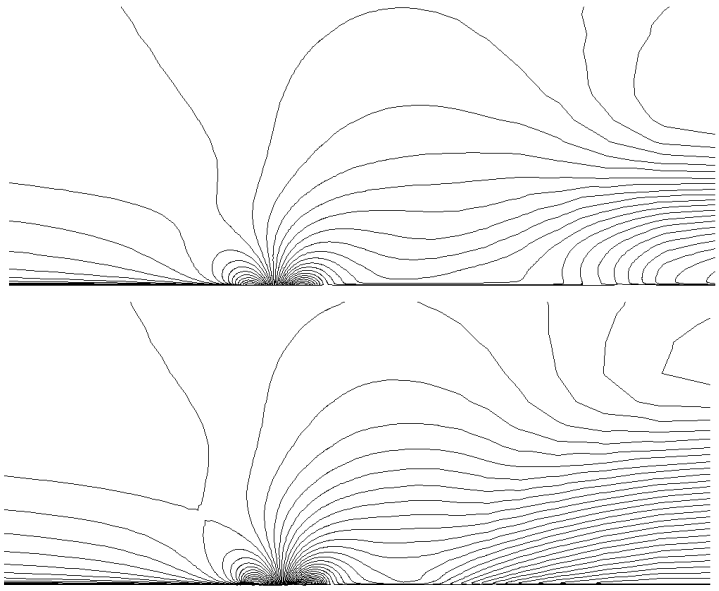


Figure 5.32: Iso-u contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 2$ (500 Hz).

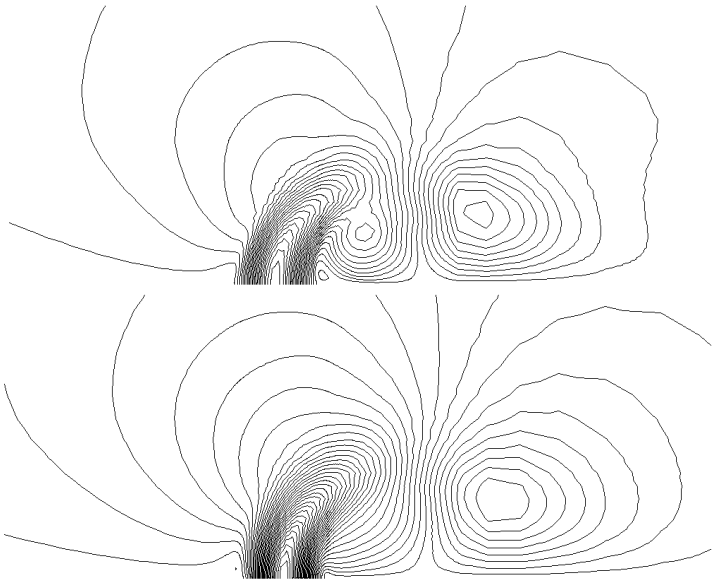


Figure 5.33: Iso-v contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 0$ (500 Hz).

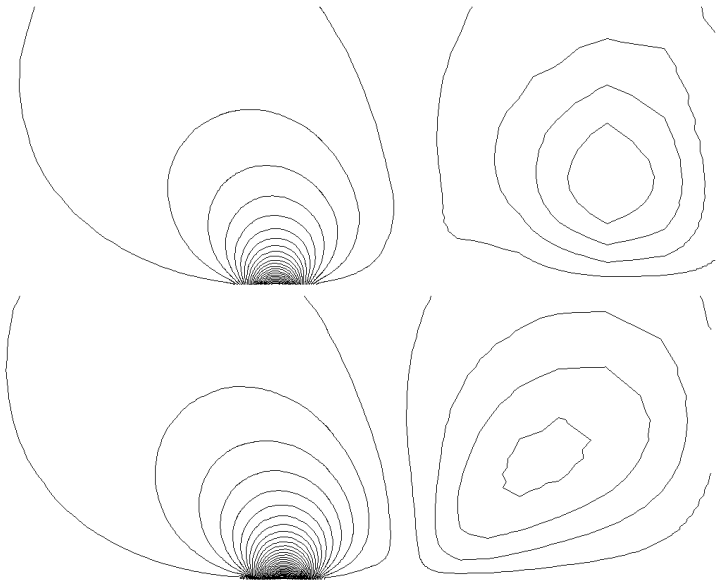


Figure 5.34: Iso-v contours for the SST k-ω (top) and Spalart-Allmaras (bottom) closures, for $\Phi = 2$ (500 Hz).

5.5 Synthesis

The objective of this study was to simulate a synthetic jet in a turbulent boundary layer flow, compare the flows predicted by some actuator models and assess the turbulence closures in this context.

It has been found that an actuator model including the slot description is a satisfactory compromise between the complexity of including the cavity and the simplicity of using only a boundary condition. The numerical parameters (convergence criterion, grid size, time step, type of boundary condition) have been set to reasonable values for the problem considered.

Regarding the influence of turbulence closures, a moderate discrepancy between the Spalart-Allmaras and the SST k-ω models has been reported, the latter model generating a more intense flow at blowing. This point will be detailed more in depth in the chapter focused on BFS case.

Part III

Optimization algorithms

Chapter 6

Context and issues

Optimization for systems governed by PDEs consists in finding the best elements in a set of solutions of a problem under some constraints. The aim is to minimize the cost function j:

$$\min_{x \in \mathcal{D}} j(x, W), \quad \text{subject to the constraints: } \psi(x, W) = 0 \tag{6.1}$$

with x the optimization variables, $\mathcal{D} \subset \mathbb{R}^n$ the search domain, W the state variables and ψ the state equations, defining implicitly the relationship $W(x)$.

Optimization in engineering is a domain that may involve several disciplines such as machine learning, data classification, partial differential equations. Optimization methods can be classified in gradient [Bonnans 2006, Wright 1999], meta-model [Jones 2001] and heuristic based methods [Michalewicz 1996] like EA, GA [Goldberg 1989], etc. The choice of the most adequate technique depends on the number of parameters to optimize, the number of allowed evaluations, the available information about the cost function derivative, convexity, etc. In the context of computational fluid dynamics, the cost function typically represents the drag, the lift or some other measurable quantities that can be accessed from the computed fields. Because of the computational burden (Euler (min), RANS (hours), URANS (days), DES (weeks), LES (months), DNS (months)), the best optimization approach would be the one that requires the minimum number of evaluations to reach optimal parameters.

Gradient based methods would thus be the most suitable. As these methods require cost function gradients with respect to the design variables, if the number of design variables is small, direct finite differences can be used. Finite differences approach has some drawbacks such as the step choice, necessity of high convergence of the simulation. When the number of variables is large, one can use adjoint methods to compute gradients. The adjoint method is very well established and has proven its benefits for steady cases in a large number of studies [Pironneau 1974, Mohammadi 2001, Jameson 2003, Dwight 2006, Brezillon 2009]. Unfortunately, adjoint techniques with unsteady flows are very cumbersome to apply as the adjoint problem has to be solved backward in time and thus necessitates to store all the states for all time steps [Belme 2011]. To reduce the cost of unsteady adjoint computations, the introduction of checkpoints [Hascoet 2013, Belme 2011] is possible to partially store the states at specific times and to recompute what is necessary. Unfortunately, this technique is still tedious to implement, and the issue of turbulence models differentiation remains problematic. Furthermore, non-linear phenomena in unsteady computations may yield multi-modality in the cost function. Direct gradient

based optimizers might get trapped in a local optimum. Thus, gradient based methods do not appear to be presently the appropriate solutions for unsteady flows optimization.

Heuristic or evolutionary techniques are out of reach because of the required number of evaluations. Meta-model based optimization appears to be a reasonable candidate and promising approach [Duvigneau 2012, Laurenceau 2008, Buche 2005], given the restriction on the number of evaluations and the small number of variables to optimize. As this optimization method is driven by the meta-model, the hard point concerns the accuracy of the model itself. Multiple possibilities are offered to constructs the meta-model. Such techniques can involve cubic spline, thin-plate spline, multiquadrics, radial basis function (RBF), artificial neural networks, etc [Jones 2001]. Most of these meta-modeling techniques have been extensively and successfully applied to machine learning problems in order to assimilate a large amount of data, to do data classification and make predictions.

The present work is focused on the Efficient Global Optimization (EGO) algorithm [Jones 1998], for reasons that will be developed in this chapter.

Chapter 7

Methods

Contents

7.1 Meta-model based algorithms (Efficient Global Optimization) . 93
7.1.1 Gaussian Process model 94
7.1.2 Correlation functions 97
7.1.3 Hyperparameters determination in Bayesian framework 98
7.1.4 Prediction with noisy observations 99
7.1.5 Construction of the Gaussian Process meta-model 101
7.1.6 Merit functions 101
7.2 Evolution strategy 103

In this chapter, the Efficient Global Optimization (EGO) method is presented, then the Evolution Strategy (ES) algorithm is detailed and used to solve meta-model training sub-problems.

7.1 Meta-model based algorithms (Efficient Global Optimization)

The *Efficient Global Optimization (EGO)* is a global optimization algorithm that takes advantage of a stochastic process model to drive the optimization [Jones 1998, Jones 2001]. As a first design of experiment phase, an initial database covering the bounded search space is generated from simulations. This database contains obviously the cost function values, but may also gather a variance estimated for each evaluation. A stochastic model, more specifically a Gaussian Process [Gibbs 1997, MacKay 1998, Williams 1998, Picheny 2011] or kriging-based model [Huang 2006, Duvigneau 2012, Duchaine 2009, Ginsbourger 2009], is constructed using this database, which allows to predict the cost function value in term of expectancy and variance at any point of the search domain. According to these predictions, the most interesting points are selected by means of a merit function. The merit function allows at the same time to identify the minimum of the function and to explore unknown regions. Once evaluated, the corresponding cost function values and the error estimations are added to the database. This process is repeated until convergence, as illustrated in Fig. 7.1.

The EGO algorithm is summarized as:

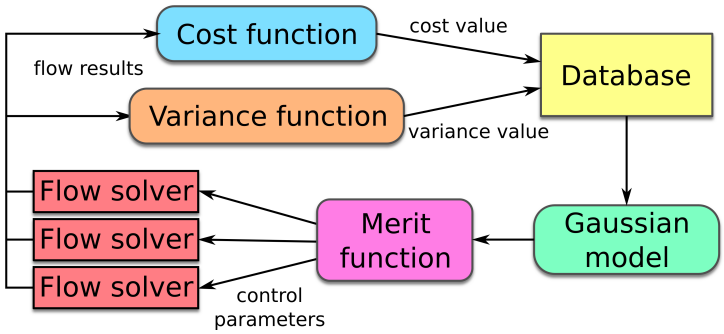


Figure 7.1: Efficient Global Optimization loop.

1. Build an *a priori* database
2. Construct a kriging or Gaussian process model
3. For $i = 1$ to p : Find the point $x_i^\star$ that minimizes the merit function to select p most interesting points
4. Evaluate the p points $(x_i^\star)_{i=1,...,p}$ and add them in the database
5. Return to step 2 until convergence

The following describes the two key elements used in the EGO algorithm: the Gaussian Process model and the merit functions.

7.1.1 Gaussian Process model

Gaussian processes are a subset of Bayesian inference methods. In statistical analysis and frequentist view, the probability of an event occurring is a number based on the frequency measure of the event itself. The interpretation of probability takes a new whole definition in the Bayesian view. The Bayesian point of view interprets a probability as a measure of belief, or confidence instead of as a measure of event occurrence.

The Bayesian approach can seem to be much more natural than the frequentists approach as it mimics the way humans think and make inferences. In his thesis, Mark N. Gibbs [Gibbs 1997] gives an introduction detailing decision making and model selection process when new evidences are raised up the ones after the others. Depending on the experience of the people, they might select one model or another given a limited amount of information, making experience very valuable.

Bayesian inference methods are able to update beliefs, or the shape of an initial probability distribution, when considering new evidences. Instead of returning a single number, or probability, Bayesian methods usually return a probability distribution.

Let n random variables $y_1, ..., y_n$ having a joint probability $p(\boldsymbol{y}) = p(y_1 ..., y_n)$. Let us split the random variables into two sets A and B with a non null union. Both sets

contains at least one element. Two groups $\boldsymbol{y_A} \in A$ and $\boldsymbol{y_B} \in B$ can be extracted. The resulting joint probability is $p(\boldsymbol{y}) = p(\boldsymbol{y_A}, \boldsymbol{y_B})$, with $\boldsymbol{y} = \boldsymbol{y_A} \bigcup \boldsymbol{y_B}$. The groups of variables $\boldsymbol{y_A}$ and $\boldsymbol{y_B}$ both contain at least one element.

To incorporate *prior* knowledge or belief $p(\boldsymbol{y_A})$ on the element A and determine a new belief called the *posterior probability* denoted by $p(\boldsymbol{y_A}|\boldsymbol{y_B})$ giving the probability of the event A occurring knowing B, the Bayes' theorem is used:

$$\underbrace{p(\boldsymbol{y_A}|\boldsymbol{y_B})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y_B}|\boldsymbol{y_A})}^{\text{likelihood}}\overbrace{p(\boldsymbol{y_A})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y_B})}_{\text{evidence}}}$$

This theorem simply comes from writing the definition of the *conditional probabilities* of $p(\boldsymbol{y_A}|\boldsymbol{y_B})$ and $p(\boldsymbol{y_B}|\boldsymbol{y_A})$:

$$p(\boldsymbol{y_A}|\boldsymbol{y_B}) = \frac{p(\boldsymbol{y_A}, \boldsymbol{y_B})}{p(\boldsymbol{y_B})}, \quad p(\boldsymbol{y_B}|\boldsymbol{y_A}) = \frac{p(\boldsymbol{y_B}, \boldsymbol{y_A})}{p(\boldsymbol{y_A})}$$

The Bayes' theorem is extensively used to tune model parameters as in this thesis or to select models given prior knowledge. Unfortunately, some terms such as the likelihood might be not be analytically tractable and one needs to use computational methods.

Gaussian process models [Rasmussen 2006] belong to response surface methods, that allow to predict a function value f at a given point x, on the basis of a set of observed values $\boldsymbol{F_N} = \{f_1, f_2, \ldots, f_N\}$ at some points $\boldsymbol{X_N} = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$. $\boldsymbol{F_N}$ is assumed to be one realization of a multivariate Gaussian process which has a joint Gaussian distribution

$$p(\boldsymbol{F_N}|C_N, \boldsymbol{X_N}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{F_N}^\top C_N^{-1}\boldsymbol{F_N}\right)}{\sqrt{(2\pi)^N \det(C_N)}}, \tag{7.1}$$

for any collection of inputs $\boldsymbol{X_N}$. C_N is the $N \times N$ covariance matrix, whose elements C_{mn} give the correlation between the function values f_m and f_n obtained at points x_m and x_n. We assume that these values are correlated, since they correspond to underlying physical phenomena. This is expressed in terms of a correlation function k, i.e., $C_{mn}(x_m, x_n; \Theta) = cov(f_m, f_n) = k(x_m, x_n; \Theta)$ with Θ a set of hyper-parameters that are supposed to be known in the following.

Now, we suppose that we would like to evaluate the function value at a new point x_{N+1}. From Eq. (7.1), the $(N+1)$-variable Gaussian process has a joint probability density of

$$p(\boldsymbol{F_{N+1}}|C_{N+1}, \boldsymbol{X_{N+1}}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{F_{N+1}}^\top C_{N+1}^{-1}\boldsymbol{F_{N+1}}\right)}{\sqrt{(2\pi)^{N+1} \det(C_{N+1})}}. \tag{7.2}$$

To predict the unknown function value f_{n+1} given the data $\boldsymbol{F_N}$, one uses the conditional

probabilities rule $p(A|B) = p(A,B)/p(B)$:

$$\begin{aligned} p(f_{N+1}|(\boldsymbol{X_N}, \boldsymbol{F_N}), C_{N+1}, x_{N+1}) &= \frac{p(\boldsymbol{F_{N+1}}|C_{N+1}, \boldsymbol{X_{N+1}})}{p(\boldsymbol{F_N}|C_N, \boldsymbol{X_N})} \\ &= \frac{\exp\left(-\frac{1}{2}(\boldsymbol{F_{N+1}}^\top C_{N+1}^{-1}\boldsymbol{F_{N+1}} - \boldsymbol{F_N}^\top C_N^{-1}\boldsymbol{F_N})\right)}{\frac{\sqrt{(2\pi)^{N+1}\det(C_{N+1})}}{\sqrt{(2\pi)^N\det(C_N)}}}. \end{aligned} \tag{7.3}$$

To simplify the expression, the covariance matrix C_{N+1} can be related to C_N by:

$$C_{N+1} = \begin{bmatrix} C_N & \boldsymbol{k_{N+1}} \\ \boldsymbol{k_{N+1}}^T & \kappa \end{bmatrix} \tag{7.4}$$

with $\kappa = k(x_{N+1}, x_{N+1}; \Theta)$ and $\boldsymbol{k_{N+1}} = [k(x_1, x_{N+1}; \Theta), \ldots, k(x_N, x_{N+1}; \Theta)]^T$.

With this relation, the inverse C_{N+1}^{-1} can be rewritten in terms of C_N and C_N^{-1}

$$C_{N+1}^{-1} = \begin{bmatrix} M & \boldsymbol{m} \\ \boldsymbol{m}^T & \mu \end{bmatrix} \tag{7.5}$$

with $M = C_N^{-1} + \frac{1}{\mu}\boldsymbol{m}\boldsymbol{m}^T$, $\boldsymbol{m} = -\mu C_N^{-1}\boldsymbol{k_{N+1}}$ and $\mu = (\kappa - \boldsymbol{k_{N+1}}^T C_N^{-1}\boldsymbol{k_{N+1}})^{-1}$.

It is important to note that at this point, only one inversion of the matrix C_N is necessary to compute C_{N+1}^{-1}.

The determinant $\det(C_{N+1})$ can be expressed as

$$\begin{aligned} \det(C_{N+1}) &= \det(C_N)\det(\kappa - \boldsymbol{k_{N+1}}^T C_N^{-1}\boldsymbol{k_{N+1}}) \\ &= \det(C_N)\frac{1}{\mu}. \end{aligned} \tag{7.6}$$

By substituting 7.5 and 7.6 in 7.3 the probability density for the function value at the new point is

$$p(f_{N+1}|(\boldsymbol{X_N}, \boldsymbol{F_N}), C_N, x_{N+1}) = \frac{1}{\hat{\sigma}_{f_{N+1}}\sqrt{2\pi}} \exp\left[-\frac{(f_{N+1} - \hat{f}_{N+1})^2}{2\hat{\sigma}_{f_{N+1}}^2}\right],$$

where

$$\hat{f}_{N+1} = \boldsymbol{k_{N+1}}^\top C_N^{-1}\boldsymbol{F_N}, \qquad \hat{\sigma}_{f_{N+1}}^2 = \kappa - \boldsymbol{k_{N+1}}^\top C_N^{-1}\boldsymbol{k_{N+1}}. \tag{7.7}$$

Thus, the probability density for the function value at the new point x_{N+1} is also Gaussian with mean $\hat{f}_{N+1}$ and standard deviation $\hat{\sigma}_{f_{N+1}}$. Therefore, the most likely value at the new point x_{N+1} is $\hat{f}_{N+1}$. This value will be considered as the prediction of the kriging model. The variance $\hat{\sigma}_{f_{N+1}}^2$ can be interpreted as a measure of uncertainty in the value prediction. The function value can be expected to vary in some range like $[\hat{f} - 3\hat{\sigma}, \hat{f} + 3\hat{\sigma}]$. Fig. 7.2 represents such an estimation at a new point with an illustration of the probability density function.

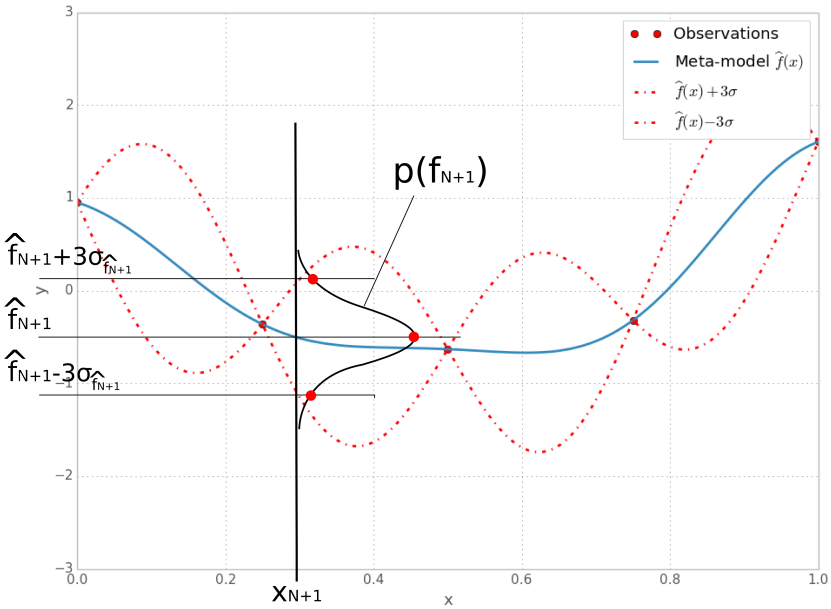


Figure 7.2: Illustration of the probability density estimation for a new point using the Gaussian Process method.

7.1.2 Correlation functions

The choice of the correlation function k is critical for the model, as it contains all the prior hypotheses, including for example regularity, symmetry, periodicity, etc. The correlation function must *a priori* reflect the characteristics of the output of the computer code.

We use here the Matérn class [Stein 1999] of covariance stationary kernels, which gives a family of correlation functions of different smoothness from the exponential covariance function to the squared exponential function. The Matérn function is defined by:

$$\mathcal{M}_\nu(x, x'; r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{|x-x'|}{r}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{|x-x'|}{r}\right) \tag{7.8}$$

with ν, r some positive parameter and K_ν the modified Bessel function of the second kind. With specific values of $\nu = p+1/2$, the Matérn function can be expressed in simple forms:

$$\begin{aligned}
\mathcal{M}_{\nu=1/2}(x, x'; r) &= e^{-\frac{|x-x'|}{r}} \\
\mathcal{M}_{\nu=3/2}(x, x'; r) &= \left(1 + \sqrt{3}\frac{|x-x'|}{r}\right) e^{-\sqrt{3}\frac{|x-x'|}{r}} \\
\mathcal{M}_{\nu=5/2}(x, x'; r) &= \left(1 + \sqrt{3}\frac{|x-x'|}{r} + \frac{5}{3}\left(\frac{|x-x'|}{r}\right)^2\right) e^{-\sqrt{5}\frac{|x-x'|}{r}} \\
\mathcal{M}_{\nu\to\infty}(x, x'; r) &= e^{-\left(\frac{|x-x'|}{r}\right)^2}
\end{aligned} \tag{7.9}$$

By using the Matérn covariance functions of parameter ν, the resulting Gaussian process is mean-squared differentiable at order k $\Leftrightarrow \nu > k$. A process $Y(x)$ is said to be mean-squared differentiable with the mean-square derivative $Y'(x)$ if the following relation holds:

$$\lim_{\varepsilon \to 0} \mathbb{E}\left[\left(\frac{Y(x+\varepsilon) - Y(x)}{\varepsilon} - Y'(x)\right)^2\right] = 0.$$

Thus with $\nu = 1/2$ the response process is continuous; with $\nu = 3/2$ it is once differentiable; with $\nu = 5/2$ it is twice differentiable; with $\nu \to \infty$ it is infinitely differentiable. For more information about continuity and differentiability of processes, the reader is referred to specialized documents [Rasmussen 2006, Diggle 2007, Stein 1999].

Combining covariance functions is possible to define new kernels. For example, to extend the Matérn function $\mathcal{M}_{\nu\to\infty}(x, x'; \theta_i)$ to its multidimensional counterpart, the following correlation function is employed:

$$k(\boldsymbol{x}, \boldsymbol{x'}; \boldsymbol{\Theta}) = \theta_1 \prod_{i=1}^{d} e^{-\left(\frac{|\boldsymbol{x}-\boldsymbol{x'}|}{r_i}\right)^2} + \theta_2, \tag{7.10}$$

with $\boldsymbol{\Theta} = \{\theta_1, \theta_2, r_1, r_2, ..., r_d\}$ the set of hyper-parameters to be determined. Practical studies have shown that the use of a non-isotropic scaling is beneficial for the problems considered here. The parameter θ_1 scales this correlation. In the second term, θ_2 gives an offset of the function values from zero. The parameters r_i represent the anisotropic distance at which the correlation decays.

7.1.3 Hyperparameters determination in Bayesian framework

As previously said, the Bayesian framework allows to make inference with prior knowledge. This framework is used here to determine the hyper-parameters Θ in the correlation function.

The constructed model must be as consistent as possible with the observed data. To do so, the following probability has to be computed:

$$p(f_{N+1}|(\boldsymbol{X_{N+1}}, \boldsymbol{F_N}), C_{N+1}) = \int p(f_{N+1}|(\boldsymbol{X_{N+1}}, \boldsymbol{F_N}), C_N, \Theta) p(\Theta|(\boldsymbol{F_N}, \boldsymbol{X_N}), C_N) d\Theta \tag{7.11}$$

This integral is analytically intractable, but a method called the *evidence maximization* [MacKay 1992] is applicable to approach it. Some other ways such as the Monte-carlo integral approximation can be also used but are described here as only the former one has been applied.

The evidence maximization makes the assumption that the posterior distribution of Θ, $p(\Theta|(\boldsymbol{F_N}, \boldsymbol{X_N}), C_N)$, is sharply peaked around a maximum Θ_{max} compared to $p(f_{N+1}|(\boldsymbol{X_{N+1}}, \boldsymbol{F_N}), C_N, \Theta)$. The evidence maximization consists in finding the most

probable set of parameters Θ_m that maximizes the posterior distribution of Θ given by:

$$p(\Theta|(\boldsymbol{X_N}, \boldsymbol{F_N}), C_N)$$

Using Bayes' theorem, this distribution can be rewritten as:

$$p(\Theta|(\boldsymbol{X_N}, \boldsymbol{F_N}), C_N) = \frac{p(\boldsymbol{F_N}|\boldsymbol{X_N}, C_N, \Theta)p(\Theta)}{p(\boldsymbol{F_N}|\boldsymbol{X_N}, C_N)}$$

Because the interest is to optimize the set of parameters Θ, the denominator is ignored as it is independent with respect to Θ. Maximizing the likelihood times the prior is the same as maximizing its log:

$$\log(p(\boldsymbol{F_N}|\boldsymbol{X_N}, C_N, \Theta)) + \log(p(\Theta)) \tag{7.12}$$

At this point, one has to give a description of the prior $p(\Theta)$. The prior $p(\Theta)$ helps to avoid solutions that would not be acceptable. If there are no prior knowledge, this term has a uniform distribution and can be neglected [Gibbs 1997]. The function to maximize $\mathfrak{L}$ becomes:

$$\mathfrak{L} = \log(p(\boldsymbol{F_N}|\boldsymbol{X_N}, C_N, \Theta)) = -\frac{1}{2}\boldsymbol{F_N}^\top C_N^{-1}\boldsymbol{F_N} - \frac{1}{2}\log\det(C_N) - \frac{N}{2}\log 2\pi. \tag{7.13}$$

Therefore, hyper-parameters are finally determined by maximizing the log-likelihood function $\mathfrak{L}$.

This function is known to be multi-modal; hence an evolution strategy that has the capability to avoid local minima, is employed for this work and described below.

7.1.4 Prediction with noisy observations

In real-life problems, either in experiments or numerical simulations, observations made on the process are noisy. In case of experiments, the noise related to gauge or instruments can usually be well estimated. In numerical simulations, errors arise from models, discretization, iterative solvers, human errors, etc. As a consequence, in both cases, we need to take into account noisy observations in our models.

According to Ginsbourger [Ginsbourger 2009], assuming a Gaussian distributed noise $\boldsymbol{\varepsilon}$, the process observation can be written as:

$$f_\varepsilon(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{\varepsilon},$$

with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \Delta)$, and $\Delta = \mathrm{diag}(\tau_i^2, i \in [0, n])$, where the error at the observation point i is thus considered as a centered Gaussian distribution with a variance τ_i^2.

By introducing the noise in the process, the covariance matrix C_N is changed [Ginsbourger 2009] to $C_N + \Delta$. As a result, the prediction of mean $\hat{f}_{N+1}$ and variance $\hat{\sigma}^2_{f_{N+1}}$ in Eq. (7.7) need to be modified to:

$$\hat{f}_{N+1} = \boldsymbol{k_{N+1}}^\top (C_N^{-1} + \Delta)\boldsymbol{F_N}, \qquad \hat{\sigma}^2_{f_{N+1}} = \kappa - \boldsymbol{k_{N+1}}^\top (C_N^{-1} + \Delta)\boldsymbol{k_{N+1}}. \tag{7.14}$$

In addition, the log-likelihood from Eq. 7.13 to be maximized becomes:

$$\mathfrak{L} = -\frac{1}{2}\boldsymbol{F_N}^T(C_N + \Delta)^{-1}\boldsymbol{F_N} - \frac{1}{2}\log(\det(C_N + \Delta)). \tag{7.15}$$

Using this methodology, the resulting meta-model is non-interpolating with non-null standard deviation at the observed points as shown on Fig.7.3.

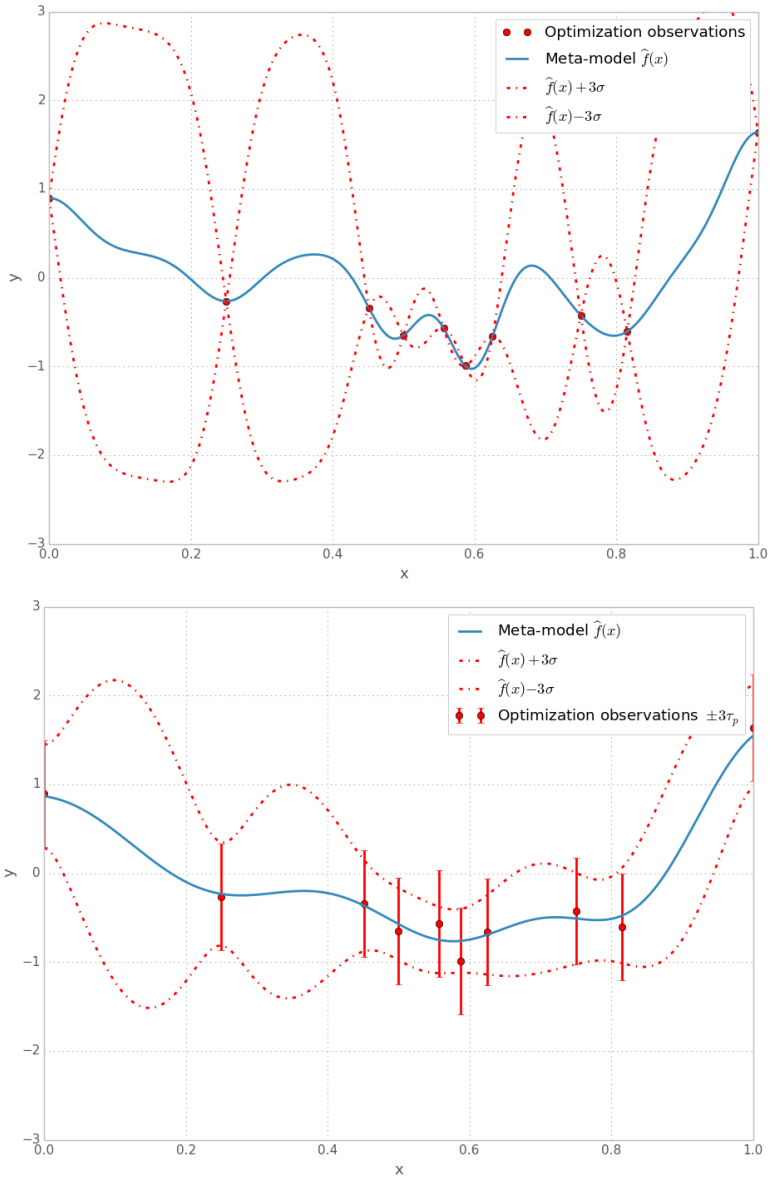


Figure 7.3: Illustration of interpolating (top) and non-interpolating (bottom) metamodels built upon 10 noisy observations. The observation error is set to $\tau_i = 0.2$.

The meta-model can now be built.

7.1.5 Construction of the Gaussian Process meta-model

All the parts to build the meta-model have been previously described. The construction of the meta-model follows the following algorithm:

1. A correlation function is selected. This function allows to build the covariance matrix when requested.
2. An initial set of hyper-parameters is selected.
3. The correlation matrix C_N based on the set of parameters and on all the available data points is computed.
4. The vector $\boldsymbol{F_N}$ is computed.
5. The system $C_N X = \boldsymbol{F_N}$ is solved using a LU decomposition of $C_N = \boldsymbol{LU}$. It allows to find $X = C_N^{-1}\boldsymbol{F_N}$.
6. The log determinant of C_N is computed using the LU decomposition: $\log(\det(C_N) = \sum_{i=0}^{N}(\log \boldsymbol{L}_{ii} + \log \boldsymbol{U}_{ii})$
7. The likelihood is evaluated using X and $\log(\det(C_N)$.
8. The likelihood is maximized by selecting new set of hyper-parameters and going to point 3 until convergence.
9. The best hyper-parameters maximizing the likelihood are fixed and used as parameters of the meta-model.
10. The meta-model is trained and can be used for prediction.

The next step is to find the most interesting points to evaluate on this meta-model. To do so, merit functions are introduced and are detailed below.

7.1.6 Merit functions

Once the model is constructed, it is used to locate the most interesting set of design parameters and values of the cost function. The naive choice of the minimum of the model may yield to non-optimum or even non-stationary points [Jones 2001]. Therefore, the new point to be evaluated is chosen by minimizing or maximizing a merit function, which utilizes the variance information given by the model. A comprehensive review of the different merit functions to locate the optimum for deterministic functions is given by Jones [Jones 2001]. Four different merit functions are used in this study, which uses the variance of the model to reach the global optimum.

Lower bound (LB): The simplest criterion is the minimization of the lower bound. This merit function is defined as:

$$LB(x) = \hat{f}(\boldsymbol{x}) - \rho\hat{\sigma}(\boldsymbol{x}) \tag{7.16}$$

where ρ is a constant typically chosen to be equal to 0, 1 or 2. Note that when $\rho = 0$, the criterion reduces to the minimization of the meta-model only.

Probability of Improvement (PI): Maximizing the probability of improvement consists in choosing a point which maximizes the probability of obtaining a value better than a given target. To do so, a target value is set, such that $T < f_{min}$ with $f_{min} = \min_{1 \leq i \leq N} f(\boldsymbol{x_i})$. This merit function is finally computed by:

$$PI(\boldsymbol{x}) = \Phi\left(\frac{T - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right), \tag{7.17}$$

where $\Phi(y) = \frac{1}{2}(1 + \mathrm{erf}(y/\sqrt{2}))$ is the cumulative distribution function of the standard normal distribution. As for LB, several target values T can be chosen.

Expected Improvement (EI): The most popular strategy is the maximization of the Expected Improvement (EI) [Jones 1998]. This approach consists in locating the point $\boldsymbol{x}$ at which the maximum reduction of the cost function can be expected. The improvement obtained by evaluating Y at the point $\boldsymbol{x}$ is defined as:

$$I := \max(f_{\min} - Y(\boldsymbol{x}), 0). \tag{7.18}$$

If we assume that the function value is the realization of a random variable with mean $\hat{f}(\boldsymbol{x})$ and standard deviation $\hat{\sigma}(\boldsymbol{x})$, the probability of reducing its value by an amount I is given by:

$$\frac{1}{\sqrt{2\pi}\hat{\sigma}(\boldsymbol{x})} \exp\left[-\frac{(f_{min} - I - \hat{f}(\boldsymbol{x}))^2}{2\hat{\sigma}(\boldsymbol{x})^2}\right]. \tag{7.19}$$

Then, the EI criterion is the expected value of I:

$$EI(\boldsymbol{x}) = \int_{I=0}^{I=\infty} I \left\{\frac{1}{\sqrt{2\pi}\hat{\sigma}(\boldsymbol{x})} \exp\left[-\frac{(f_{min} - I - \hat{f}(\boldsymbol{x}))^2}{2\hat{\sigma}(\boldsymbol{x})^2}\right]\right\} dI. \tag{7.20}$$

After an integration by part, it is shown that:

$$EI(\boldsymbol{x}) = \hat{\sigma}(\boldsymbol{x})[u\Phi(u) + \phi(u)], \quad u(\boldsymbol{x}) = \frac{f_{min} - \hat{f}(\boldsymbol{x})}{\hat{\sigma}(\boldsymbol{x})}, \tag{7.21}$$

where $\phi(y) = \frac{1}{\sqrt{2\pi}} \exp(-y^2/2)$ is the standard normal distribution.

Note that a recent result on convergence proof of the EI method under fixed mean and covariance functions is given in [Vazquez 2010].

Augmented Expected Improvement (AEI): The main drawback of the EI criterion is that the minimum f_{min} is not exactly known in the presence of noise. The Augmented Expected Improvement [Huang 2006] is based on two modifications in case of noisy functions. First, instead of using the minimum f_{min}, it uses a surrogate, addressing the problem that f_{min} is not exactly known. The used surrogate is $T = \hat{f}(\boldsymbol{x}^{**})$ with $\boldsymbol{x}^{**}$ (*effective best solution*) the minimum of $\hat{f} + \hat{\sigma}$ over the over the current database. Then, a

penalty term is introduced in order to take into account the error for the next evaluation τ_{N+1}. This penalty enhances exploration by penalizing points with small variance $\hat{\sigma}^2$ with respect to noise τ_{N+1}^2. Finally, the AEI criterion becomes:

$$\begin{aligned} AEI(\boldsymbol{x}) &= EI_T(\boldsymbol{x}) \times \left(1 - \frac{\tau_{N+1}}{\sqrt{\hat{\sigma}(\boldsymbol{x})^2 + \tau_{N+1}^2}}\right), \\ EI_T(\boldsymbol{x}) &= \hat{\sigma}(\boldsymbol{x})[u\Phi(u) + \phi(u)], \\ u(\boldsymbol{x}) &= \frac{T - \hat{f}(\boldsymbol{x})}{\hat{\sigma}(\boldsymbol{x})} \end{aligned} \tag{7.22}$$

The AEI criterion has exhibited the best results for the optimization of noisy functions among many merit functions [Picheny 2011].

All of these merit functions are minimized using the evolution strategy algorithm, which is detailed in the following.

7.2 Evolution strategy

Evolution strategy (ES) algorithms are used to calibrate or train the hyper-parameters of the Gaussian Process model and to minimize merit functions. Evolution strategy algorithms are meta-heuristic optimization methods inspired by natural evolution. Its first description goes back to the 1960s in Germany by I. Rechenberg and H.-P. Schwefel and it was applied in a numerical context by D. Hartmann. This optimization technique is similar to Differential Evolution, Genetic Algorithm, etc., in the sense that they all belong to the field of *evolutionary algorithms*. These algorithms can reach global optimum and their cost to reach convergence is usually in the order of some thousands of evaluations. They are based on iterative evolution of a population, in which individuals are characterized by a fitness value conditioned by the problem to optimize. The evolution of the population individuals is generally described by reproduction, mutation, recombination and selection operators.

In ES, an individual $\mathcal{I}$ in a population $\mathcal{P}$ is composed with a pair of floating point valued vectors, which contains information about the decision vector $\boldsymbol{x}$ and standard deviation $\boldsymbol{\sigma}$ for the mutation process:

$$\mathcal{I} = \mathcal{I}(\boldsymbol{x}, \boldsymbol{\sigma})$$

The associated mutation operator of the individual mutates $\boldsymbol{x}$ and $\boldsymbol{\sigma}$. If only the mutation of $\boldsymbol{x}$ is considered, it follows a vector of random Gaussian numbers with zero mean and standard deviations $\boldsymbol{\sigma}$ denoted by $\mathcal{N}(0, \boldsymbol{\sigma})$. Considering a parent $\mathcal{I}_p(\boldsymbol{x_p}, \boldsymbol{\sigma_p})$ that mutates into a child $\mathcal{I}_c(\boldsymbol{x_c}, \boldsymbol{\sigma_c})$, then a possible mutation operator is:

$$\mathcal{M}_{ut}(\mathcal{I}_p) = \mathcal{I}_c(\boldsymbol{x_p} + \mathcal{N}(0, \boldsymbol{\sigma_p}), \boldsymbol{\sigma_p}).$$

With this specific operator, the standard deviation never evolves. On the other side, by only considering the mutation of $\boldsymbol{\sigma}$, the common sense to improve the convergence

towards the optimum would be that the variation of $\boldsymbol{x}$ decreases with the number of generations [Michalewicz 1996]. This self-adaptation behavior can be set through the standard deviation mutation with an exponential decrease of σ:

$$\mathcal{M}_{ut}(\mathcal{I}_p) = \mathcal{I}_c(\boldsymbol{x_p}, \boldsymbol{\sigma_p} \exp(\mathcal{N}(0, \tau)).$$

with $\tau < 1$. In practice [Beyer 2002], the exponential decrease $\tau = 1/\sqrt{2N}$, with N the number of parameters to optimize.

To set the ideas of the ES algorithm, let us define a population of individuals $\mathcal{P}_p$ of size μ that are called the parents, and a population $\mathcal{P}_c$ of λ offsprings larger than μ deduced from the parents that are recombined into μ parents at the next generation. From the parent population, the mutation operator is applied to generate the λ offsprings. Then μ parents for the next generation are selected either from the population $\mathcal{P}_c$ or from the population $\mathcal{P}_p \cup \mathcal{P}_c$. These two strategies are respectively called the $(\mu, \lambda) - ES$ and $(\mu + \lambda) - ES$ [Beyer 2002].

In this work, the implemented ES algorithm is the $(\mu, \lambda) - ES$. The full algorithm is:

1. An initial starting temporary individual is specified as a starting point $\mathcal{I}_t^0(\boldsymbol{x_t^0}, \boldsymbol{\sigma_t^0})$ and stored into the parent population $\mathcal{P}_p^0$.

2. At the generation n, the temporary individual $\mathcal{I}_t^n$ is mutated by the mean of the mutation operator to create λ individuals of indices $i = 0, ..., (\lambda - 1)$:

   $$\mathcal{I}_{c_i}^n(\boldsymbol{x_{c_i}^n}, \boldsymbol{\sigma_{c_i}^n}) = \mathcal{M}_{ut}(\mathcal{I}_t^n)$$

   with $\boldsymbol{\sigma_{c_i}^n} = \boldsymbol{\sigma_t^n} \exp(\mathcal{N}(0, \tau))$ and $\boldsymbol{x_{c_i}^n} = \boldsymbol{x_t^n} + \mathcal{N}(0, \boldsymbol{\sigma_{c_i}^n})$.

   These newly created individuals are stored in the child population $\mathcal{P}_c^n$ by replacing the previous ones.

3. The first μ best individuals among $\mathcal{P}_c^n$ are selected and stored in the parent population $\mathcal{P}_p^n$.

4. The parents contained in $\mathcal{P}_p^n$ are combined to produces a temporary descendant $\mathcal{I}_t^{n+1}(\boldsymbol{x_t^{n+1}}, \boldsymbol{\sigma_t^{n+1}})$. The reproduction operator used is based on the average of the population attributes:

   $$\mathcal{I}_t^{n+1}(\boldsymbol{x_t^{n+1}}, \boldsymbol{\sigma_t^{n+1}}) = \mathcal{R}_{ep}(\mathcal{I}_{p_0}^n, \mathcal{I}_{p_1}^n, ..., \mathcal{I}_{p_{\mu-1}}^n) = \mathcal{I}_t^{n+1}\left(\frac{\sum_{i=0}^{\mu-1} \boldsymbol{x_{p_i}^n}}{\mu}, \frac{\sum_{i=0}^{\mu-1} \boldsymbol{\sigma_{p_i}^n}}{\mu}\right)$$

5. Return to step 2 until convergence.

Concerning numerical parameters, the population size for the children is set to $\lambda = 4 + 3\log(N_p))$ with N_p the number of parameters to optimize, the population size of the parents is $\mu = \lambda/2$ and the initial step is set to $\sigma_0 = 1/3$ assuming that the parameters were scaled between 0 and 1. The number of generations is set to $20N$ to train the meta-model and $100N$ to optimize the merit function.

Within this section, evolution strategy algorithms have been described. A great amount of work have been conducted in the field and a good review of such algorithms can be found in [Beyer 2002]. Due to its popularity and its applicability, the evolution strategy algorithm has been extended in various ways to correct its drawbacks. The last evolution of the evolution strategy is CMA-ES, for Covariance Matrix Adaptation, developed by Nikolaus Hansen at INRIA [Hansen 2006]. It uses local statistical information to adapt its distribution for the mutation operator .

The next chapter aims to validate the implementation of the optimization algorithms on analytical test functions.

Chapter 8

Validation

Contents

8.1 1D analytical case . . . 107
8.1.1 Deterministic case optimization . . . 107
8.1.2 Optimization with noise . . . 108
8.2 Noisy Branin function . . . 110
8.2.1 Deterministic case optimization . . . 112
8.2.2 Optimization with noise . . . 112
8.3 Conclusion on optimization validation . . . 114

To validate the meta-model based optimization for deterministic as well as noisy functions, several optimization cases are considered. This part is focused on the optimization of analytical functions with 1 and 2 parameters. The first two will help us to understand how the noise affects the convergence, the second one to have a more representative test case corresponding to the real simulation problem to optimize.

8.1 1D analytical case

The following unconstrained problem [Picheny 2011] is considered:

$$Minimize \quad f(x) = \frac{1}{2}\left(\frac{\sin(20x)}{1+x} + 3x^3\cos(5x) + 10(x-0.5)^2 - 0.6\right) + \varepsilon_p \tag{8.1}$$

with $\varepsilon_p \sim \mathcal{N}(0, \tau_p^2)$, a random perturbation with a normal distribution of mean 0 and standard deviation τ_p. The initial database contains 5 points equally distributed in the domains.

The analytical function and the repartition of the sample points are shown on the Fig. 8.1.

8.1.1 Deterministic case optimization

First the analytical function is defined without any noise with $\tau_p = 0$. Four figures among the first 8 steps of the optimization using the expected improvement merit function are

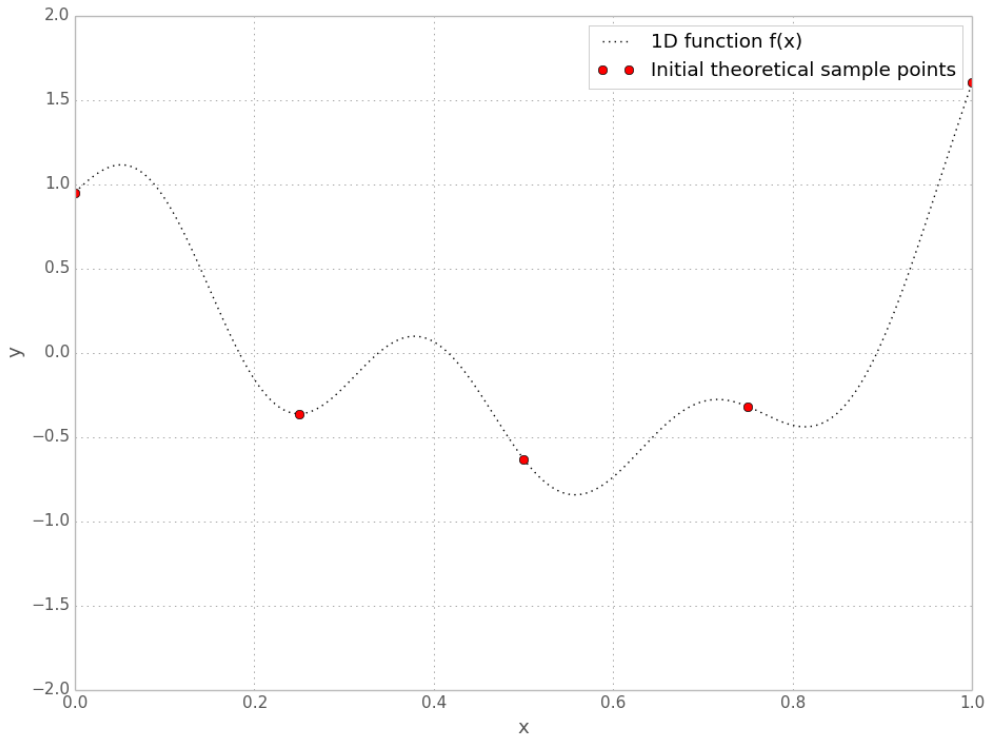


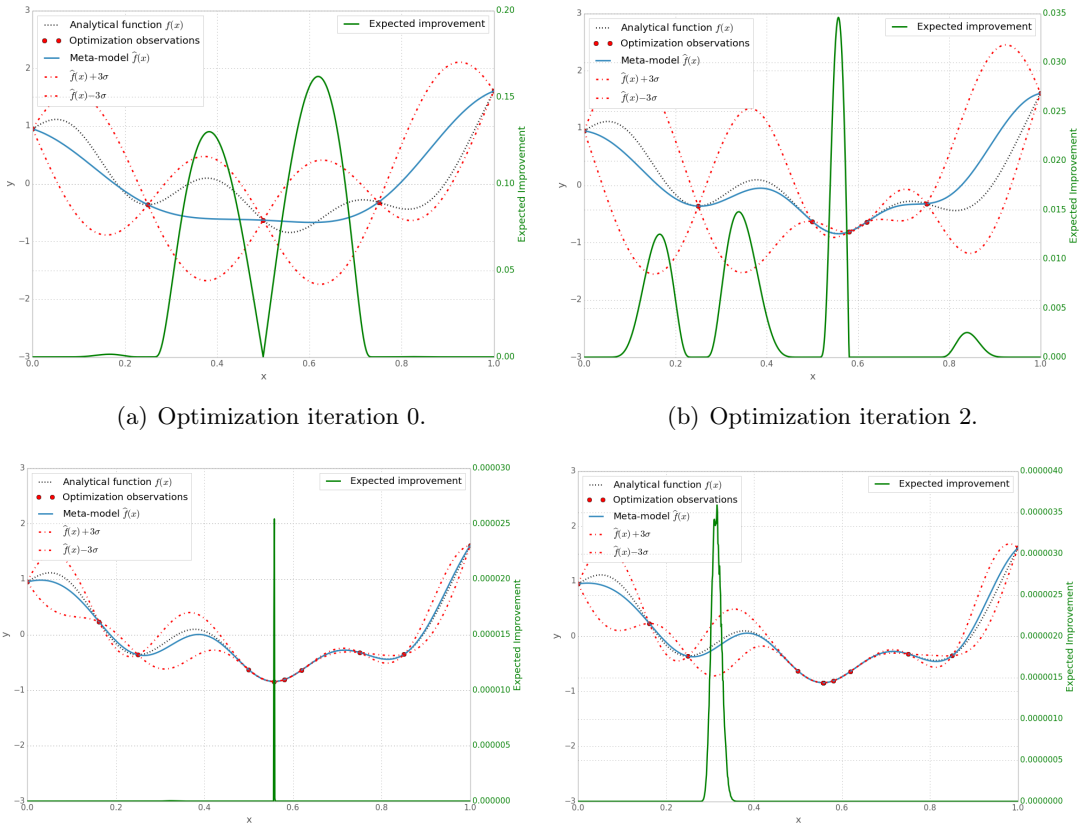
Figure 8.1: 1D analytical function and initial database.

shown on Fig. 8.2. As seen the optimization process is very fast and approximates the global optimum within the first 5 iterations. The merit function in green on the curves shows how the merit function drives the optimization. The next points to be evaluated are added where this merit function is maximum.

8.1.2 Optimization with noise

The same case is considered but with some noise on each evaluated points with $\tau_p = 0.1$. This model simulates a solver which is not able to give an exact output because of coarse convergence for example. First of all, the same algorithm using an interpolating meta-model is used, then a non interpolating one is applied. On the latter, the error variance is set to τ_p^2.

Optimization with an interpolating model: An optimization with an interpolating meta-model and an EI merit function is considered. Four figures among the first 15 steps of the optimization are shown on Fig. 8.3. As we could expect, the optimization of a noisy function using an interpolating meta-model gives a very oscillating model with a strong uncertainty in the low-sampled regions. Whatever the merit function considered, this algorithm would fail to find the global minimum in a limited number of iterations as the meta-model does not account for the noise.



(a) Optimization iteration 0.

(b) Optimization iteration 2.

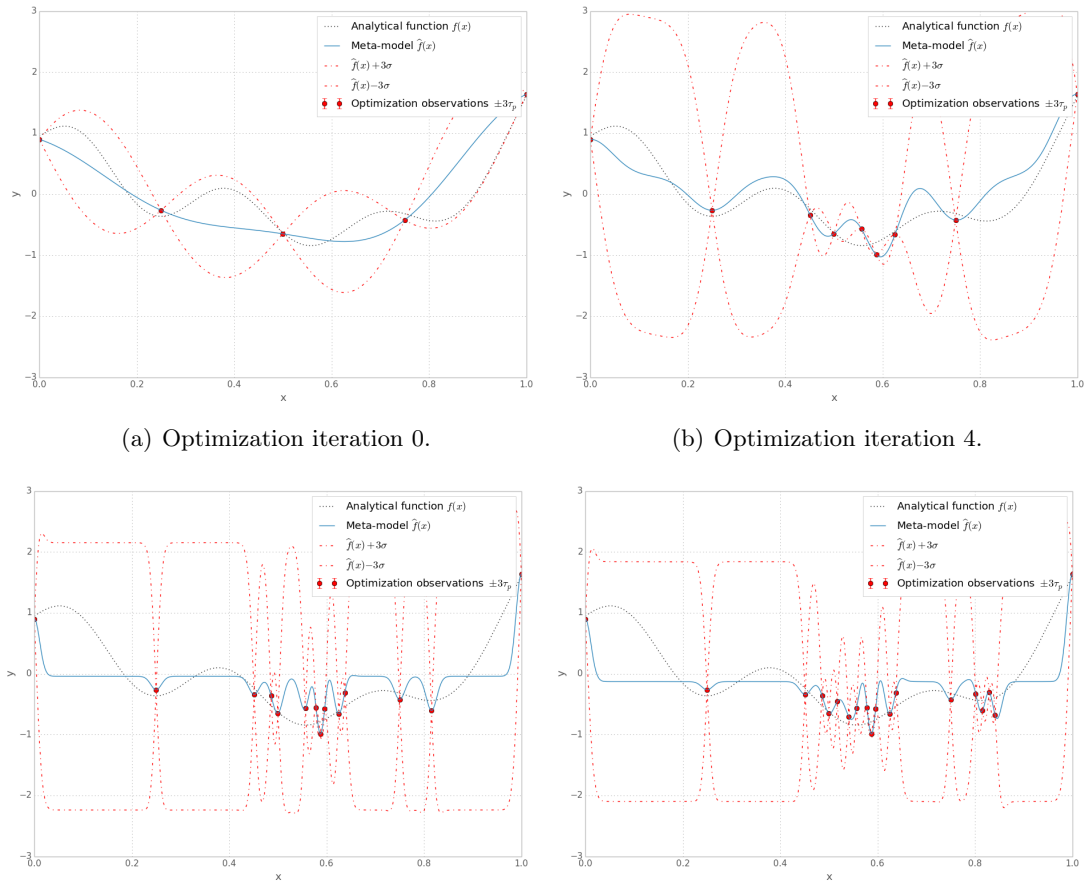
(c) Optimization iteration 5.

(d) Optimization iteration 7.

Figure 8.2: Optimization with the expected improvement of a 1D function using $\tau_p = 0$ and an interpolating meta-model. The second axis shows the expected improvement merit function. Note that the axis range is not kept constant as the expected improvement can take very dispersed values.

Optimization with a non interpolating model: As previously seen, increasing the points in the database for a noisy function yields an oscillating meta-model. To address this problem, we consider here the meta-model including noisy observations.

First, an optimization with a non-interpolating meta-model using the EI merit function is considered. Four steps among the 15 optimization iterations are shown on Fig. 8.4. These figures show that the meta-model converges to the global minimum without noise. Furthermore, as the number of iteration increases, the meta-model remains stable, which is not the case with an interpolating meta-model.



(a) Optimization iteration 0.

(b) Optimization iteration 4.

(c) Optimization iteration 9.

(d) Optimization iteration 14.

Figure 8.3: Optimization with the expected improvement of a 1D function using $\tau_p = 0.1$ and an interpolating meta-model.

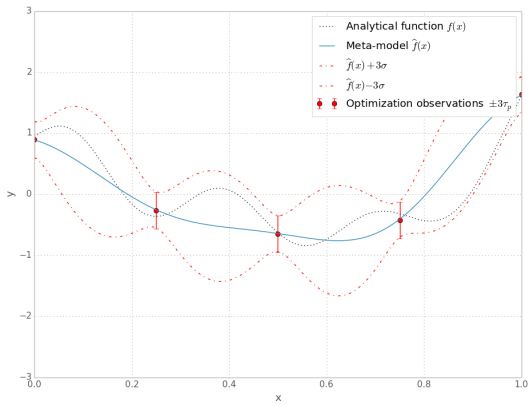
8.2 Noisy Branin function

The Branin function is a 2D test function for unconstrained global optimization [Jones 1998, Jamil 2013]. The minimization problem with additional noise term is defined by:

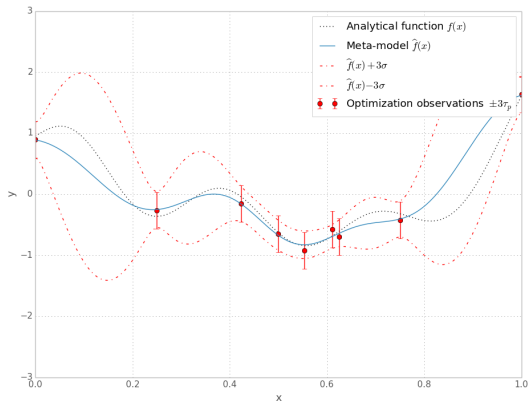
$$Minimize \quad f(x,y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10 + \varepsilon_p,$$

with $\varepsilon_p \sim \mathcal{N}(0, \tau_p^2)$, a random perturbation with a normal distribution of mean 0 and standard deviation τ_p.

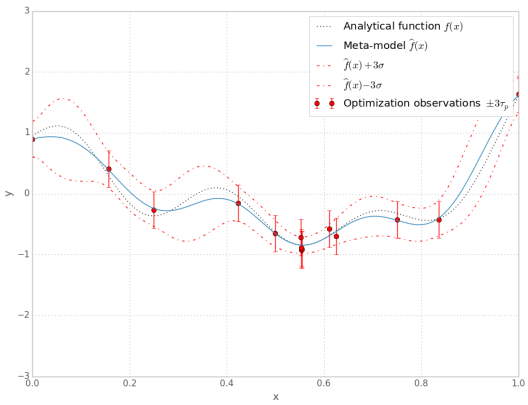
The Branin function has three global optima located at $x^* = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$, where the function $f(x^*) = 0.397887$ with $\varepsilon_p = 0$. It is represented on Fig. 8.5 The initial database contains 8 points distributed in the domains following an optimized Latin Hypercube Sampling unless specified.



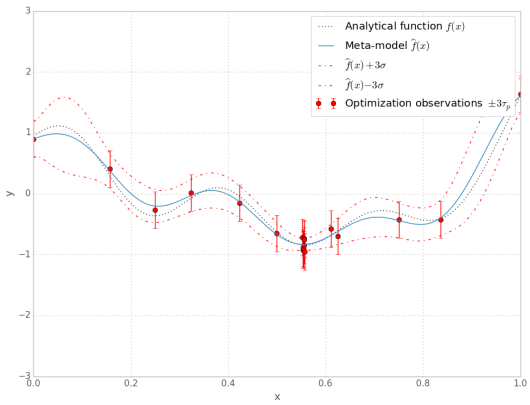
(a) Optimization iteration 0.



(b) Optimization iteration 4.



(c) Optimization iteration 9.



(d) Optimization iteration 14.

Figure 8.4: Optimization with the expected improvement of a 1D function using $\tau_p = 0.1$ and non-interpolating meta-model, with an observation uncertainty standard deviation set to $\tau_i = 0.1$.

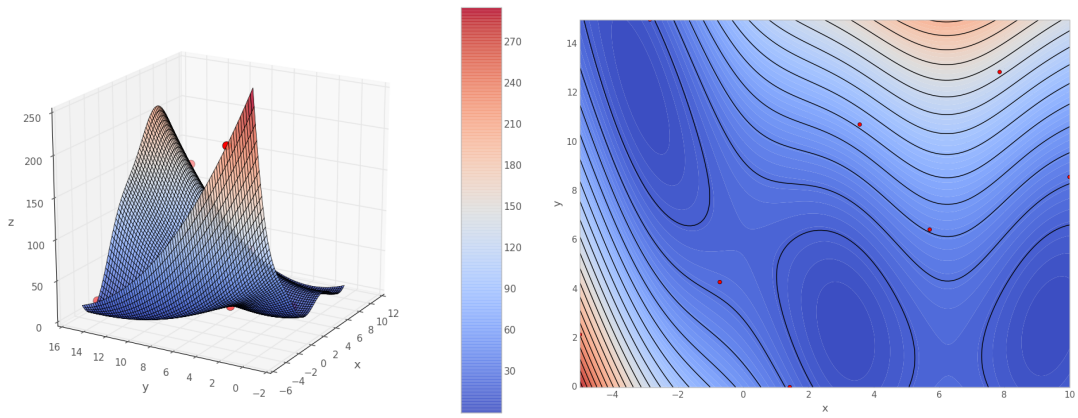
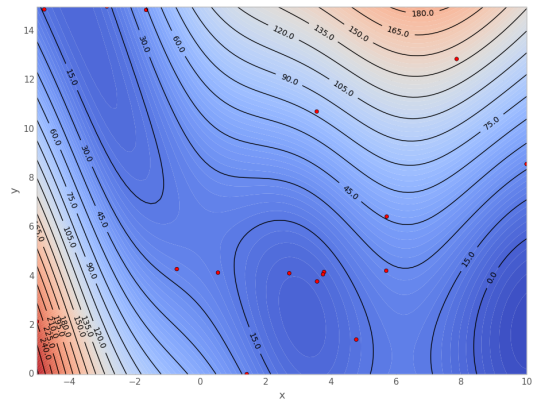


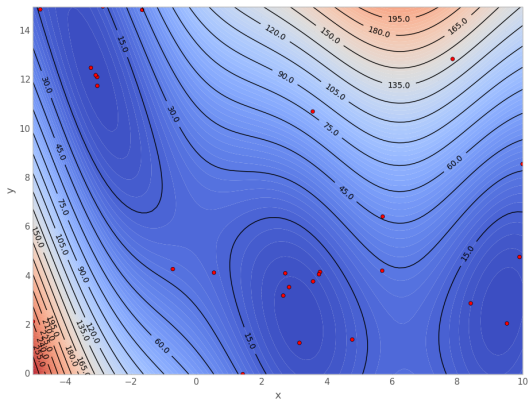
Figure 8.5: The Branin function.

8.2.1 Deterministic case optimization

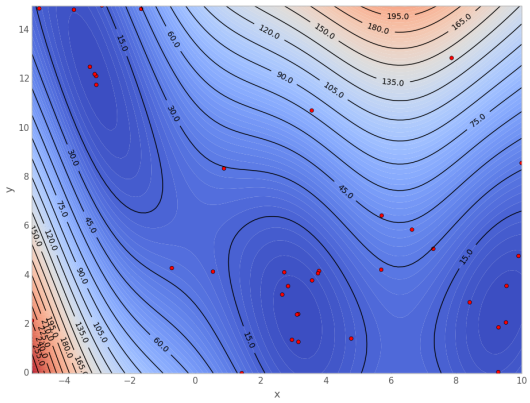
First a noise free optimization is considered ($\tau_p = 0$), with the PI and the EI merit functions. Four figures among the first 40 steps of the optimization are shown on Fig. 8.2.1 and Fig. 8.2.1. For both the PI and EI merit functions, the three optima are located



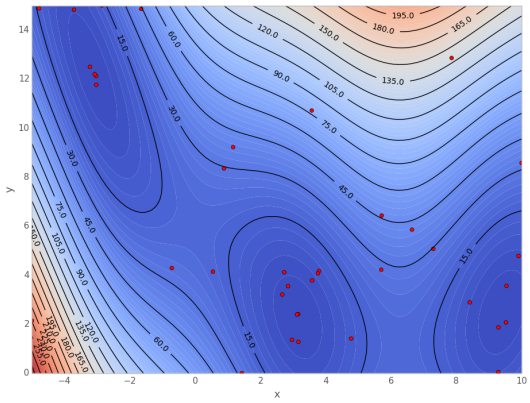
(a) Optimization iteration 10.



(b) Optimization iteration 20.



(c) Optimization iteration 30.



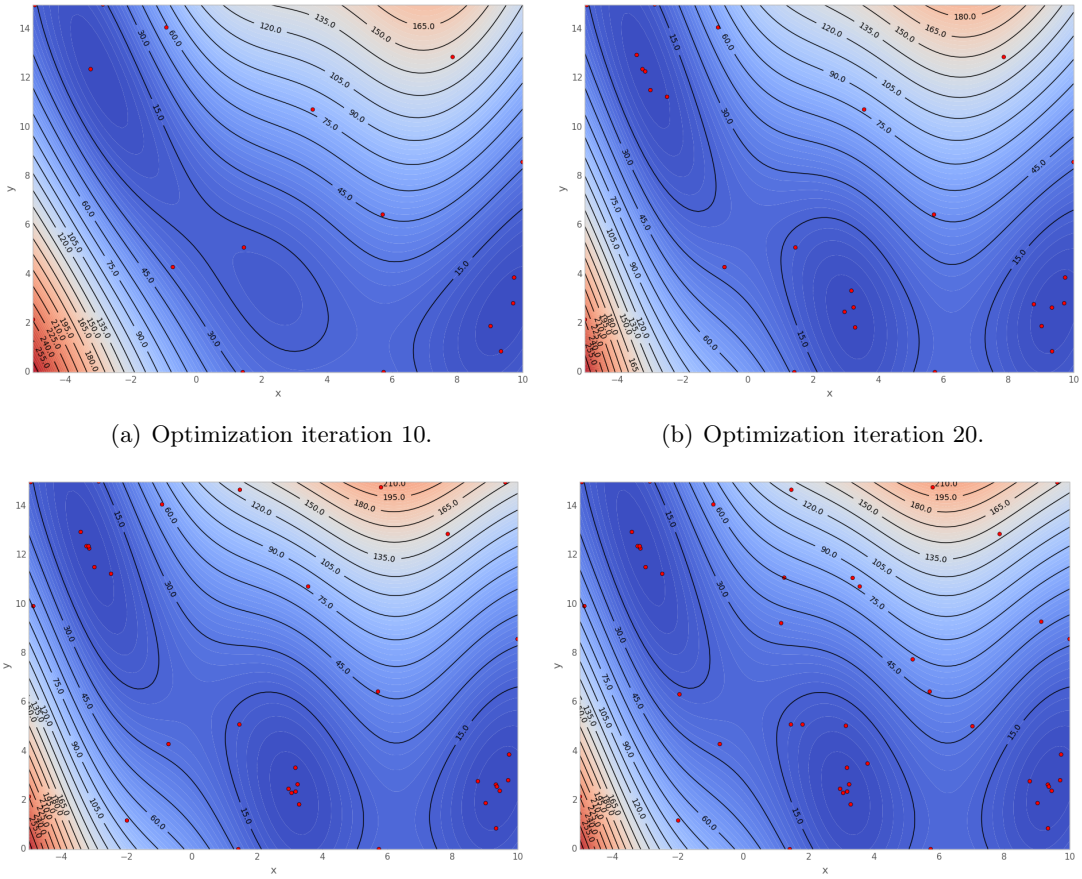
(d) Optimization iteration 40.

Figure 8.6: Optimization with the probability of improvement merit function of the Branin function using $\tau_p = 0$.

within the first 20 iterations. In both cases, the algorithm converges to the optima when the number of iterations increases. Furthermore, from the iteration 20, the algorithm converges towards the optima with no difficulty. It is seen on Fig. 8.8 that the convergence speed using EI is slightly better than the convergence using PI.

8.2.2 Optimization with noise

We consider now a the same case, but with some noise $\tau_p = 5$.



(a) Optimization iteration 10.

(b) Optimization iteration 20.

(c) Optimization iteration 30.

(d) Optimization iteration 40.

Figure 8.7: Optimization with the expected improvement merit function of the Branin function using $\tau_p = 0$.

Optimization with an interpolating model: An optimization with an interpolating meta-model and an EI merit function is first considered. Four figures among the first 20 iterations of the optimization are shown on Fig. 8.2.2. As for the 1D case, the optimization of a noisy function using an interpolating meta-model gives a very oscillating model as illustrated on the Fig. 8.2.2 for the iteration 20. The oscillations lead to convergence issues toward the optimums.

Optimization with a non interpolating model: An optimization with a non-interpolating meta-model is considered to account for noise. A first case is considered with the error variance set to τ_p^2 using LB, EI, PI and AEI merit functions. Then the error variance is set to $(\frac{\tau_p}{5})^2$, $(\frac{\tau_p}{2})^2$ and $(2\tau_p)^2$ with the EI merit function to analyze the influence of τ_p on the meta-model.

The optimization with error variance set to τ_p^2 using LB, EI, PI and AEI are respectively found on the Fig. 8.11(a), Fig. 8.11(b), Fig. 8.11(c) and Fig. 8.11(d) for the first 50 iterations. The figures show a good convergence of the algorithm towards the optima.

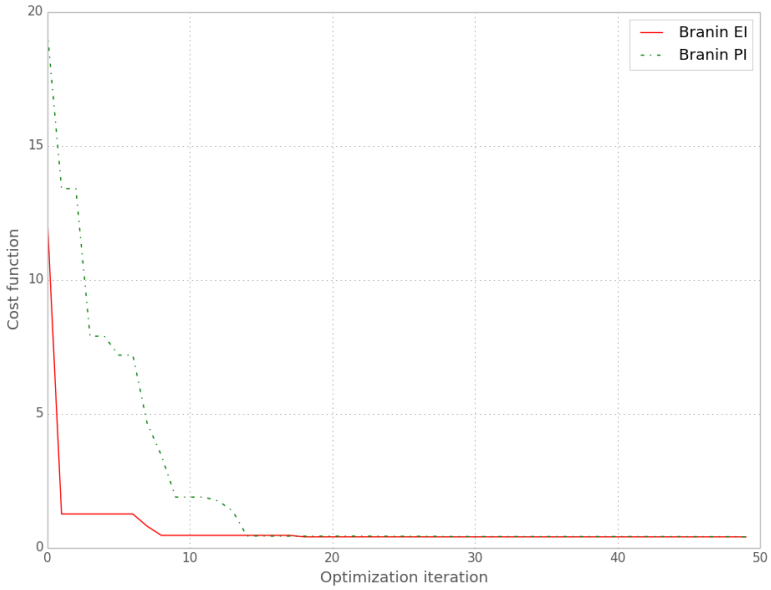


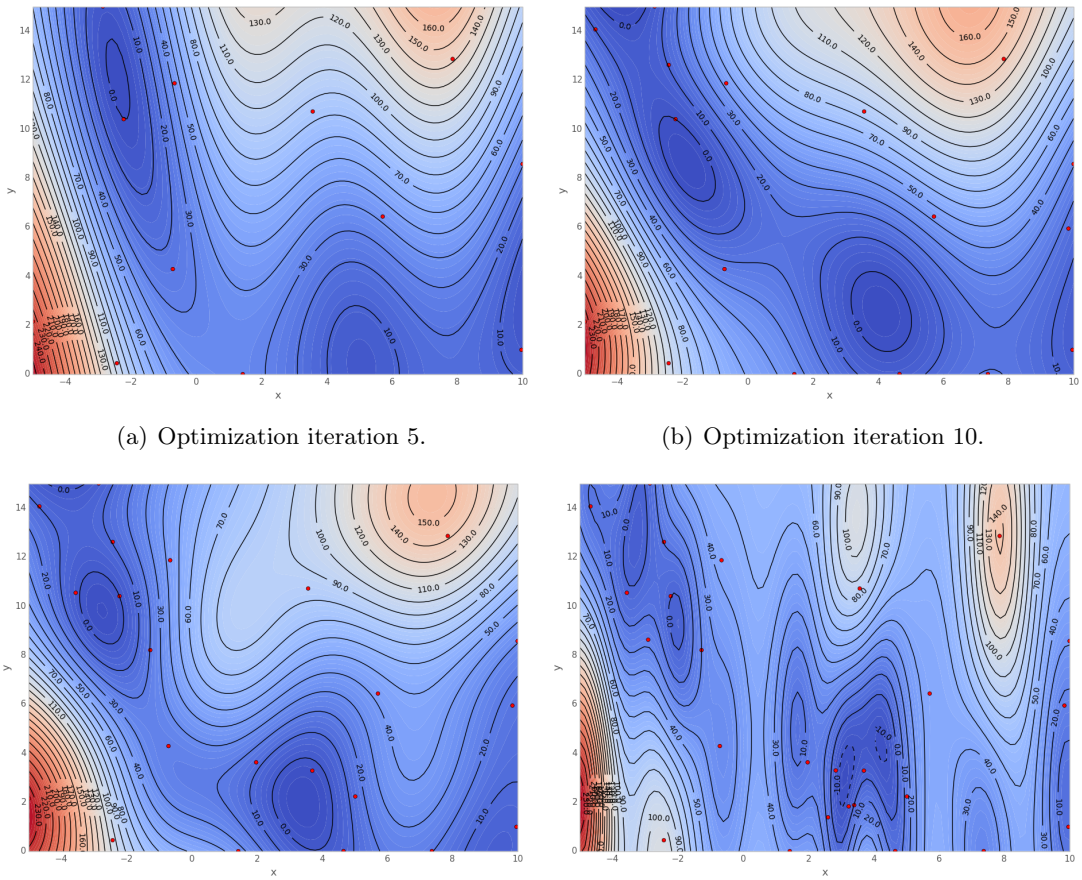
Figure 8.8: Optimization convergence for EI and PI on the Branin function.

For all merit functions, the three optima are well approached and the meta-model is stable and close to the function without noise thanks to the non-interpolating implementation of the Gaussian Process. For a complete comparative results of the merit functions on different analytical problems, we refer the reader to the work of Picheny et al. in [Picheny 2011].

Finally,the Fig. 8.12 presents the results for the optimization of the noisy Branin function with $\tau_p = 5$ and the error variance τ_i set to $(\frac{\tau_p}{5})^2$, $(\frac{\tau_p}{2})^2$, $(\tau_p)^2$ and $(2\tau_p)^2$ in order to measure the impact of noise estimation. It is clearly seen that the optimization with a too low error variance τ_i doesn't filter enough the noise and lead to an oscillatory meta-model. On the other side, if the filtering is too high, the meta-model is too much smoothed leading to a loss of accuracy in the optima locations. From this study, setting $\frac{\tau_p}{2} \leq \tau_i \leq 2\tau_p$ lead to a reasonable optimization as all the optima are found.

8.3 Conclusion on optimization validation

The validation of EGO techniques shows the importance of introducing a noise filtering control when optimizing with noisy functionals. In fact, if the noise is not considered, the optimization does not converge and the meta-model begins to oscillate through the iterations. In this part, we have shown that the implementation by modifying the covariance function of the noise in Gaussian Processes is extremely efficient for all the test cases. Furthermore, it can be noticed that 30 functions evaluations are usually enough to find at least one minimum.



(a) Optimization iteration 5.

(b) Optimization iteration 10.

(c) Optimization iteration 15.

(d) Optimization iteration 20.

Figure 8.9: Optimization with the expected improvement of the Branin function using $\tau_p = 5$ and an interpolating meta-model.

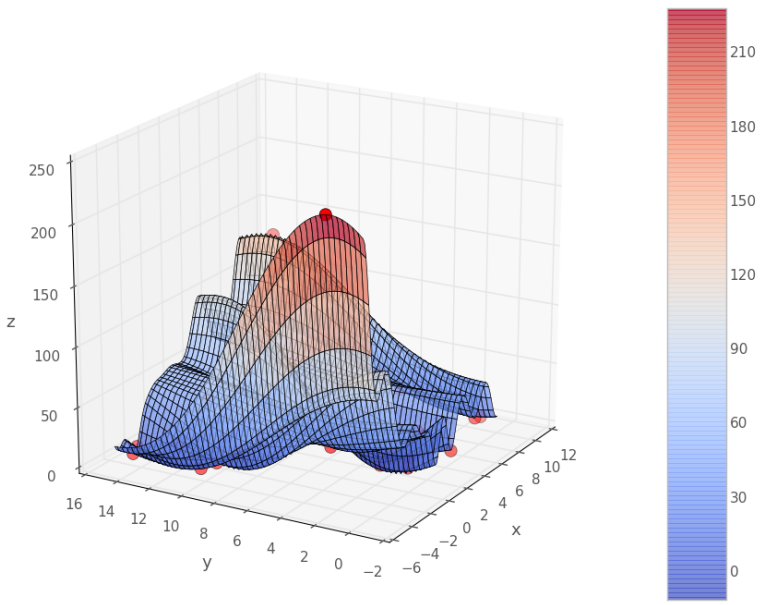
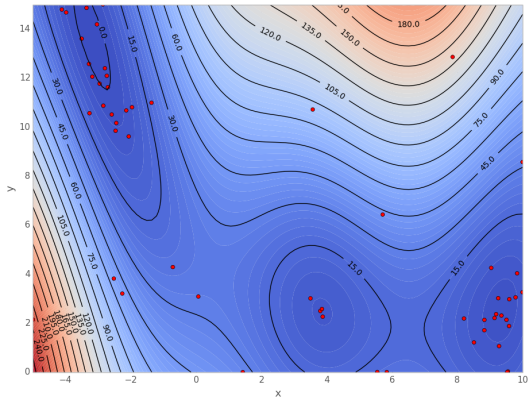
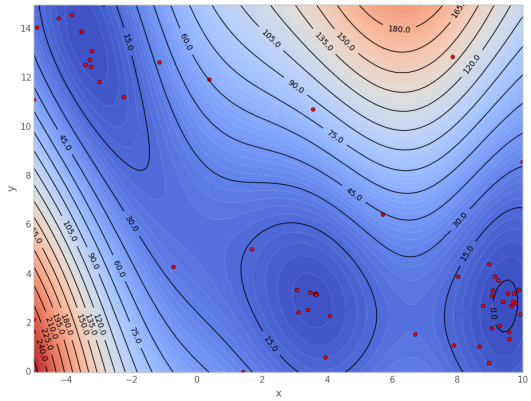


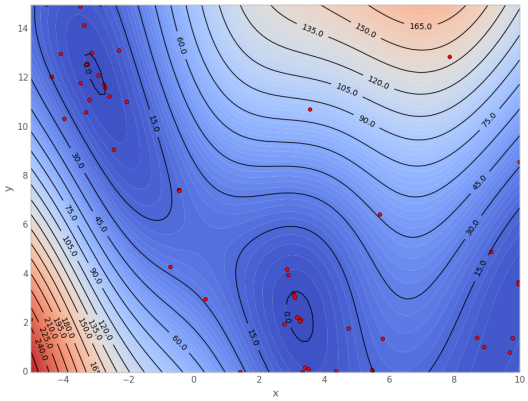
Figure 8.10: Iteration 20 for the Branin function optimization using an interpolating meta-model and an EI merit function. The meta-model is very oscillating leading to a convergence issue.



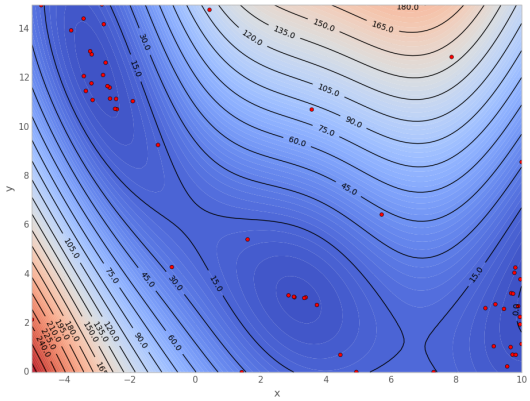
(a) Optimization with LB merit function.



(b) Optimization with EI merit function.

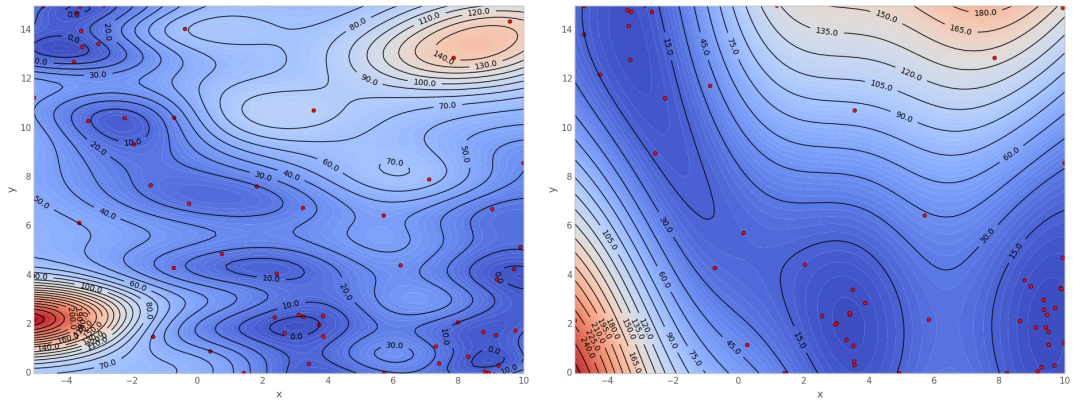


(c) Optimization with PI merit function.



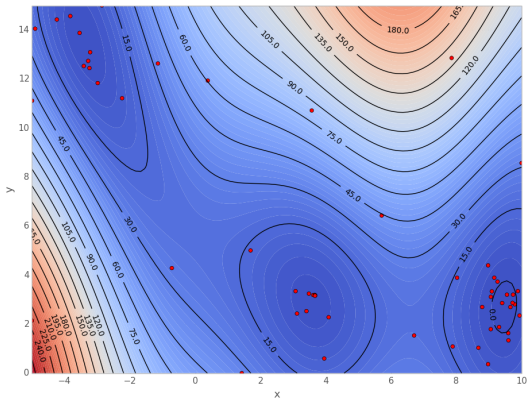
(d) Optimization with AEI merit function.

Figure 8.11: Optimization results after 50 evaluations (around 16 iterations for LB and PI) of the Branin function using $\tau_p = 5$ and the meta-model error variance set to τ_p^2.

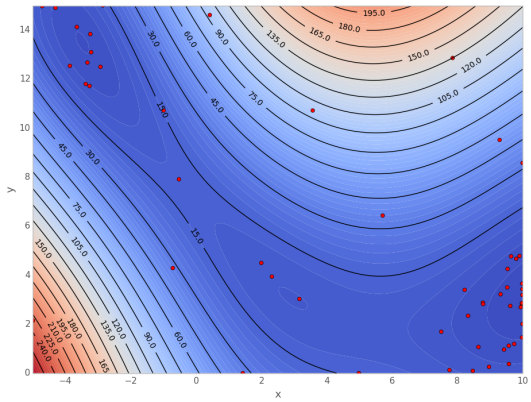


(a) Optimization with $\tau_i = \frac{\tau_p}{5}$.

(b) Optimization with $\tau_i = \frac{\tau_p}{2}$.



(c) Optimization with $\tau_i = \tau_p$.



(d) Optimization with $\tau_i = 2\tau_p$.

Figure 8.12: Optimization results after 50 evaluations of the noisy Branin function ($\tau_p = 5$) using the EI merit function. Comparison of error variance τ_i setting influence.

Part IV

Application to flow control

In the previous parts, the optimization algorithms and numerical methods to solve PDE systems have been validated with respect to analytical functions and test cases from the literature. In the context of the FP7 MARS (Manipulation of Reynolds stress for separation control and drag reduction) European project, a particular attention has been raised in the study of detached flow control. Two test cases are considered: the NACA0015 airfoil and the Backward Facing Step. In the first case, the detachment is due to the adverse pressure gradient while in the second case, the detachment is imposed by the geometry.

Chapter 9

NACA0015

Contents

9.1 Test-case configuration 123
9.2 Synthetic jet optimization 130
9.2.1 Impact of observation variance 130
9.2.2 EI/PI comparison 132
9.2.3 Analysis of optimum controlled flows 136
9.3 Discussion 153

9.1 Test-case configuration

The NACA0015 is a symmetric airfoil with a maximum thickness of 15% of the chord. Its shape is generated with the equation:

$$z = \pm\frac{t}{0.2}c\left[0.2969\sqrt{\frac{x}{c}} + (-0.1260)\left(\frac{x}{c}\right) + (-0.3516)\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 + (-0.1036)\left(\frac{x}{c}\right)^4\right] \tag{9.1}$$

with $c = 1m$ and the maximum thickness $t = 15/100$. The last coefficient is a modified coefficient (originally set to -0.1015) in order to retrieve a sharp trailing edge.

A synthetic jet with a diameter $d_{jet} = 0.5\%c$ is added to the initial shape at a location of $x_{jet} = 12\%c$. Its geometrical model is based on a slot with a height of $h_{slot} = 2\%c$.

The grid corresponding to this configuration is composed of 78186 vertices and is represented on Fig.9.1-9.2. 17 points are placed at the bottom of the jet to allow a good description of the sinusoidal shape distribution. Approximately 600 points are placed on the upper surface of the wing. The far-field is located at $100c$ from the airfoil. The distance between the first node and the wall is 10^{-5} to fulfill the $y^+ < 1$ criterion at all time steps, at all locations on the wing surface and for all closures.

The flow is considered to be fully turbulent, with a Reynolds number of $Re = 5 \times 10^6$ based on the chord and a Mach number of $M = 0.15$. The lift polars for such a configuration with Spalart-Allmaras and k-ω SST turbulence closure shown on Fig. 9.3, enhance

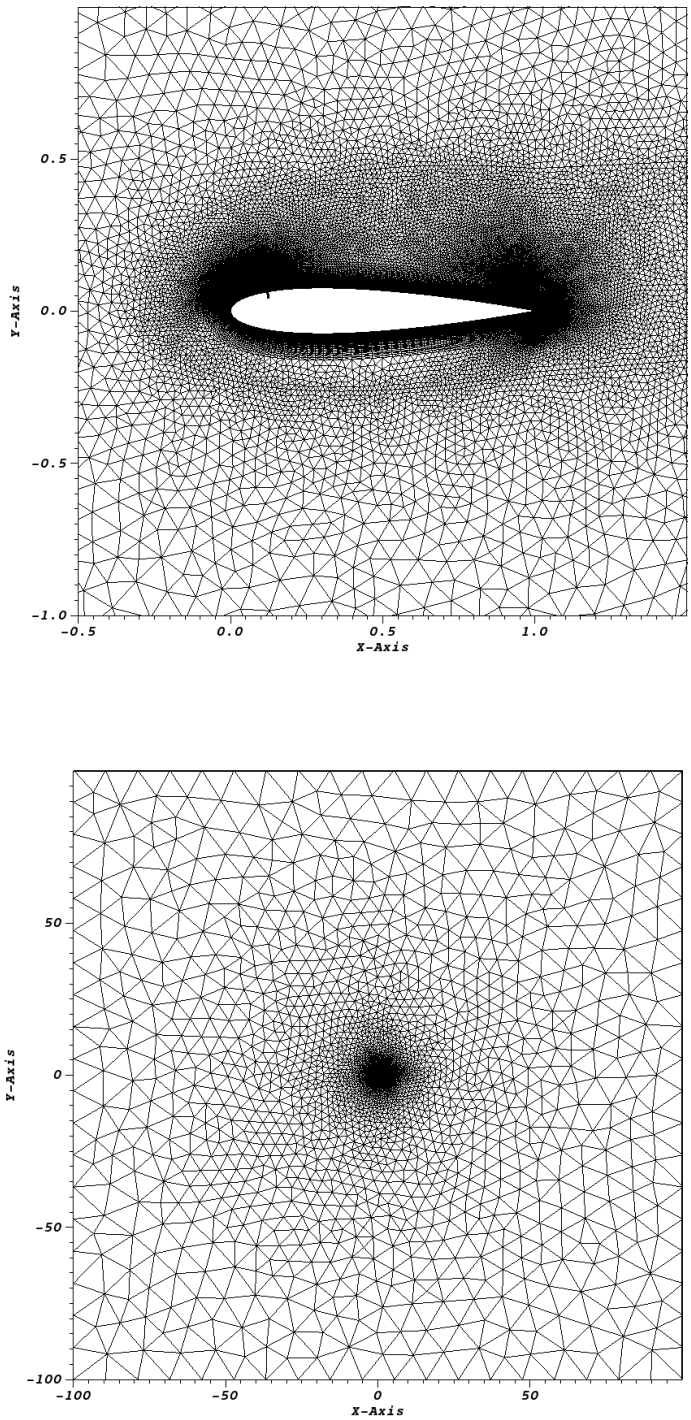


Figure 9.1: NACA0015 configuration - general view.

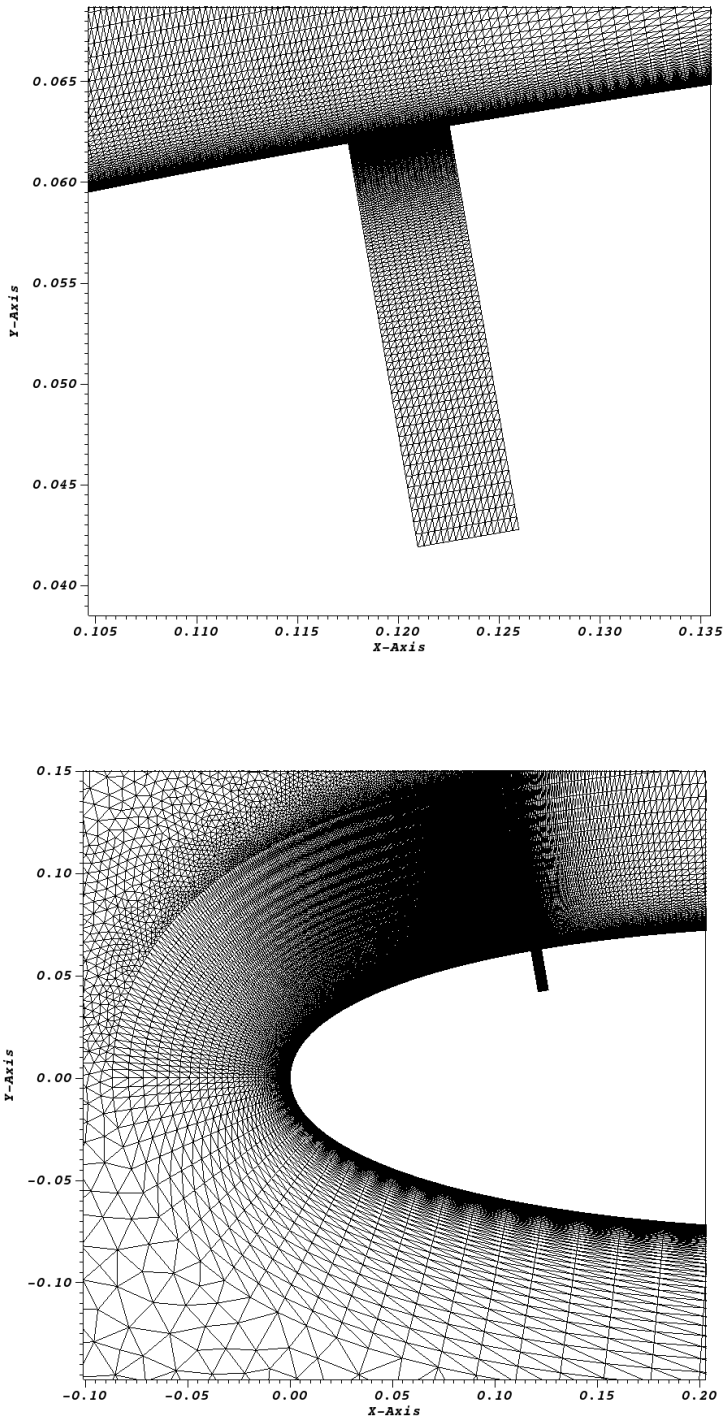


Figure 9.2: NACA0015 configuration - slot view.

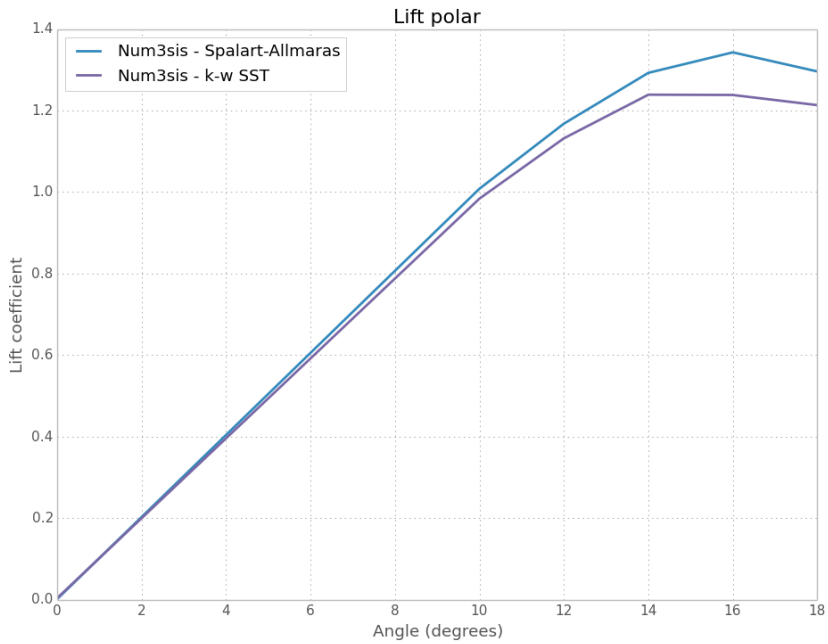


Figure 9.3: NACA0015 lift polar for the Spalart-Allmaras and k-ω SST turbulence models.

Jet parameters	Min	Max
U_{jet}/U_{inf}	0.5	2
$f_{jet}L_{ref}/U_{inf}$	0.5	2

Table 9.1: Jet parameters for the optimization over the NACA0015.

the fact that the wing is stalling after approximately 16 degrees. As a consequence, the angle of attack is set to $\alpha = 18\,\text{deg}$ to provide a detached flow with a recirculation on the suction side of the wing. The farfield boundary condition is modeled using Riemann invariants, the airfoil with adiabatic wall boundary condition and the jet with the previously described conditions. The Spalart-Allmaras and k-ω SST are selected here.

The objective of the optimization is to maximize the time averaged lift. The optimized parameters are the jet amplitude and frequency within the intervals described in Tab. 9.1 with U_{inf} the freestream velocity, and L_{ref} the airfoil chord. These intervals were chosen following the work of [Duvigneau 2006] to be certain that at least one optimum is present in the search domain.

The time averaged lift $\overline{P}$ is computed on the last synthetic jet period using the following formula:

$$\overline{P} = \cos(\alpha)\overline{F_y} - \sin(\alpha)\overline{F_x}$$

with α the flow angle of attack, $\overline{F_x}$ and $\overline{F_y}$ the time averaged aerodynamic forces projected on the x and y axis.

	Spalart-Allmaras closure	k-ω SST closure
Lift coefficient	1.34	1.26

Table 9.2: Lift coefficients for the NACA0015 at 18 degrees using the Spalart-Allmaras and k-ω SST turbulence closures.

The resulting time averaged lift coefficient $\overline{C_l}$ is found by:

$$\overline{C_l} = \frac{\overline{P}}{1/2\rho_{inf}U_{inf}^2 L_{ref}}$$

with ρ_{inf} and U_{inf} freestream density and velocity, and L_{ref} the airfoil chord.

As numerical parameters, 256 time steps per synthetic jet period and a reduction of non-linear residuals of 3 orders per time-step are selected.

An initial study of the non controlled flow around the NACA0015 airfoil is first performed at 18 deg. In this perspective, the jet boundary at the bottom of the slot is replaced by a wall boundary condition to model a non moving synthetic jet. The unsteady computation is run with a time step of $\Delta t = 0.002s$ until a quasi-stationary flow is finally found for this configuration.

Fig. 9.4 is a plot of the velocity magnitude for the Spalart-Allmaras closure. The same quantities for the k-ω SST closure are shown on Fig. 9.5. The initial configuration shows a primary large clockwise and a small secondary counterclockwise recirculation zones past the flow detachment point at $x \approx 0.5$. The behaviors of the two closures are very similar with some slight discrepancies within the recirculation area, where the k-ω SST closure exhibits a slightly larger primary recirculation zone. The lift coefficient for the closures are detailed on Tab. 9.2. The two closures present similar flow behaviors and lift coefficients. These values are taken as reference for the optimization process.

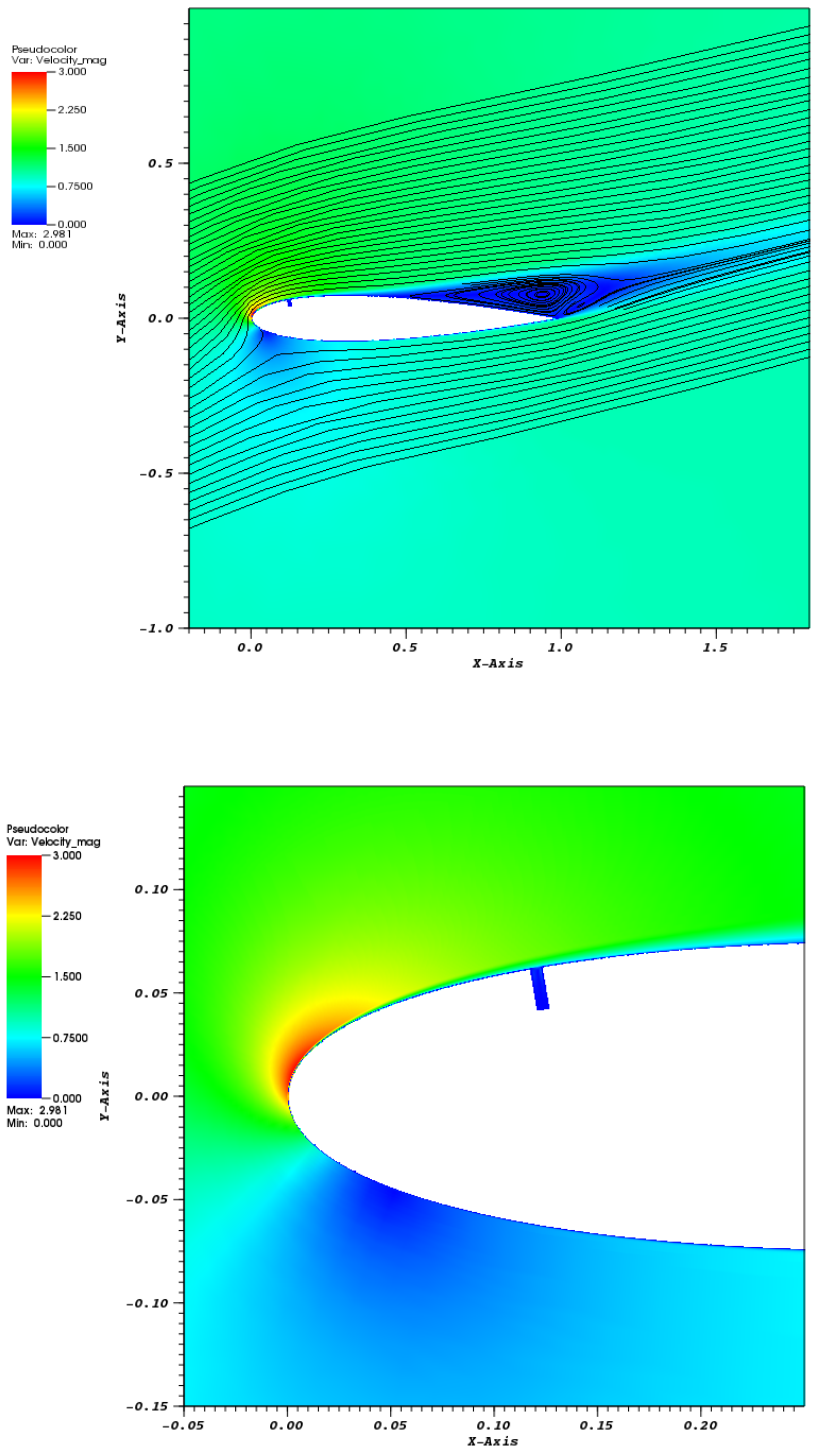


Figure 9.4: NACA0015 velocity at 18 degrees with Spalart-Allamaras turbulence closure.

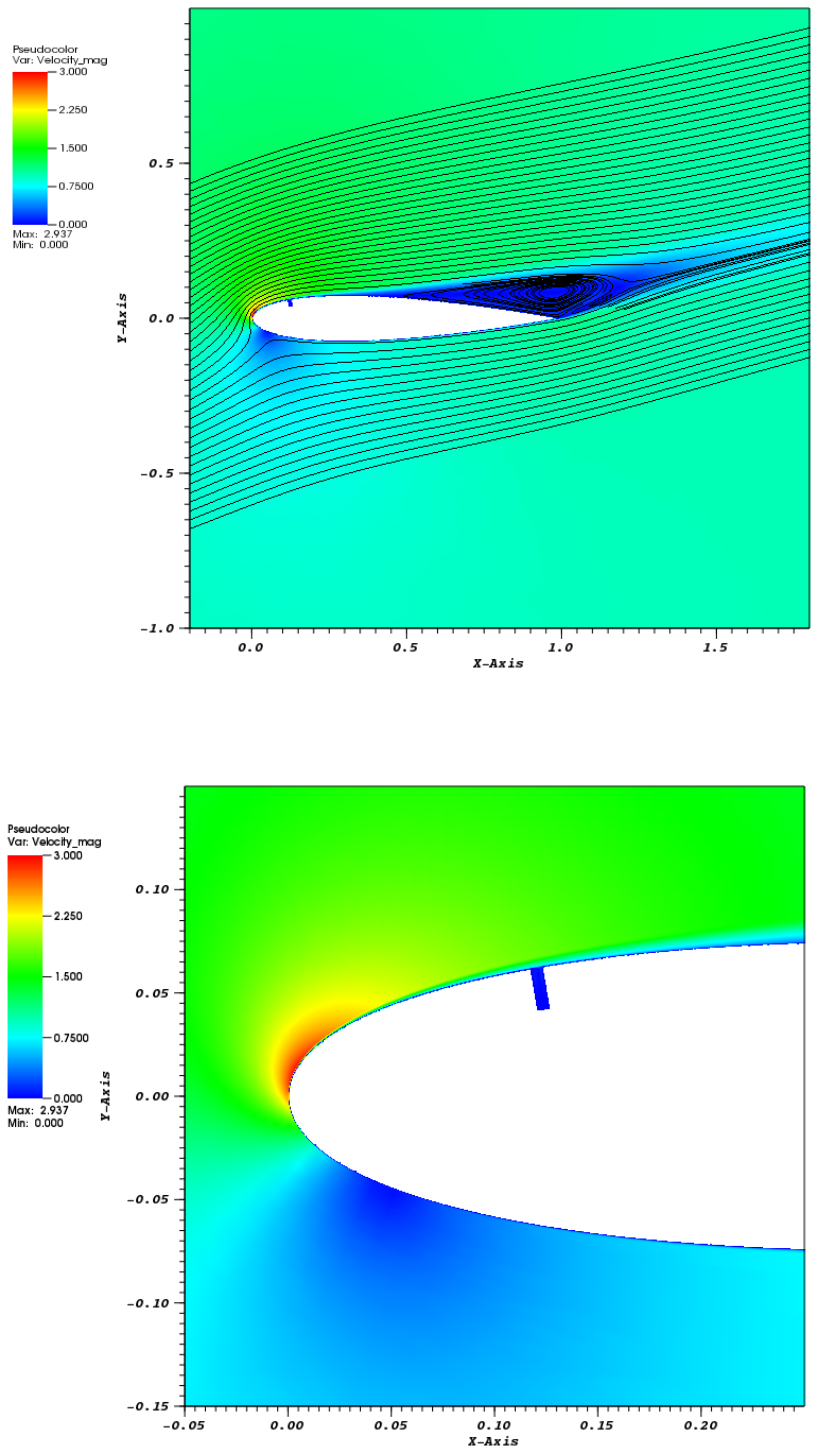


Figure 9.5: NACA0015 velocity at 18 degrees with k-ω turbulence closure.

Point id	Amplitude (adim)	Frequency (adim)
0	0.71	0.14
1	0.86	0.43
2	0.29	0.29
3	0.14	0.00
4	0.57	1.00
5	0.00	0.71
6	0.43	0.57
7	1.00	0.86

Table 9.3: Initial database sampling for the optimization using the NACA0015.

9.2 Synthetic jet optimization

The initial database for the optimization of the synthetic jet parameters is composed of 8 design points generated using a latin hypercube sampling between 0 and 1. A linear relation is used to map the optimizer design points to the actuation parameters. This database is described on Tab. 9.3 and represented on Fig. 9.6.

The initial metamodels for the Spalart-Allmaras and the k-ω SST closures are shown on Fig. 9.7.

The two meta-models differ in the mid-frequency and low amplitude. This difference is introduced by the point 2 (0.29, 0.29). This difference gives a first insight on the turbulence model impact on detached and controlled flow simulations.

9.2.1 Impact of observation variance

Based on this database, optimization using the EI and PI merit functions are performed. After the first 11th optimization iterations with an error variance of $\tau_i^2 = 0$ using EI, the metamodels assuming an error variance of $\tau_i^2 = 0$ and an error variance of $\tau_i^2 = 5 \times 10^{-3}$ are compared. The metamodels for the k-ω SST are shown on Fig. 9.8. The selected error variance of $\tau_i^2 = 5 \times 10^{-3}$ is a threshold value for which the metamodel shape sharply changes. Increasing the value of the variance then has low impact on the shape of the model. As seen on the figures, observation errors are polluting the meta-model, leading to oscillations just as for the noisy Branin function optimization. This error is likely to come from a lack of proper residuals convergence at each pseudo time steps or a proper stabilization in time of the cost function despite of the care taken for the simulations. Exactly the same behavior and error variance level holds for the Spalart-Allmaras closure.

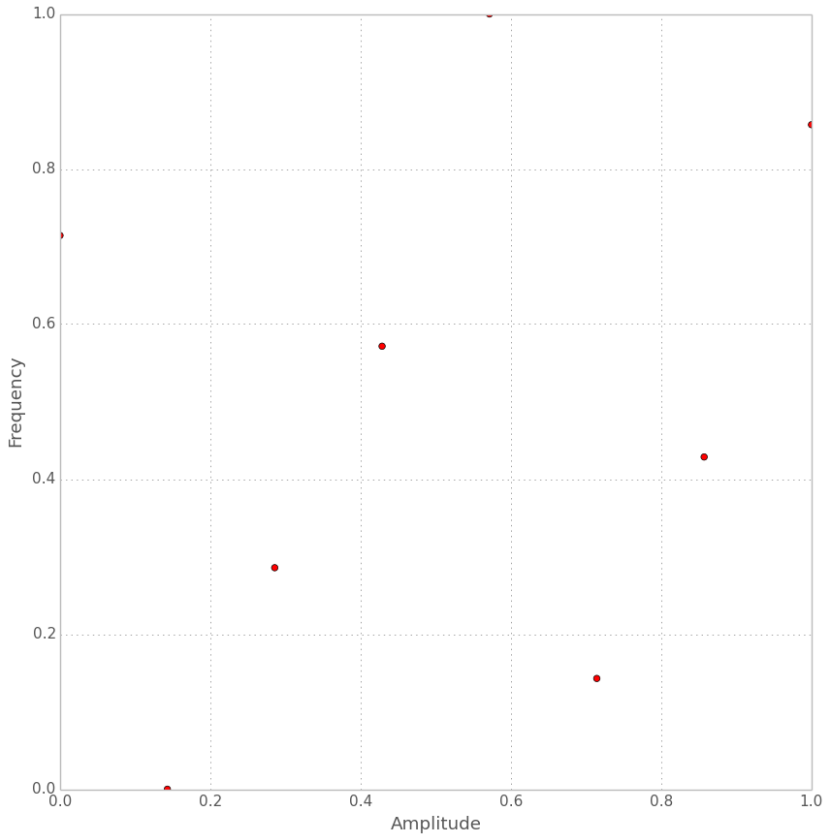


Figure 9.6: Initial database generated for the optimization of jet amplitude and frequency on the NACA0015.

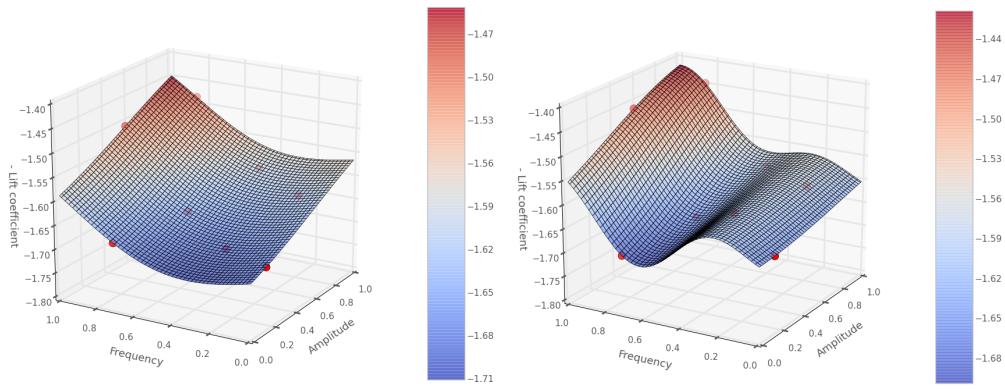


Figure 9.7: Initial database metamodels for the k-ω-SST and the Spalart-Allmaras closures for the NACA0015 configuration.

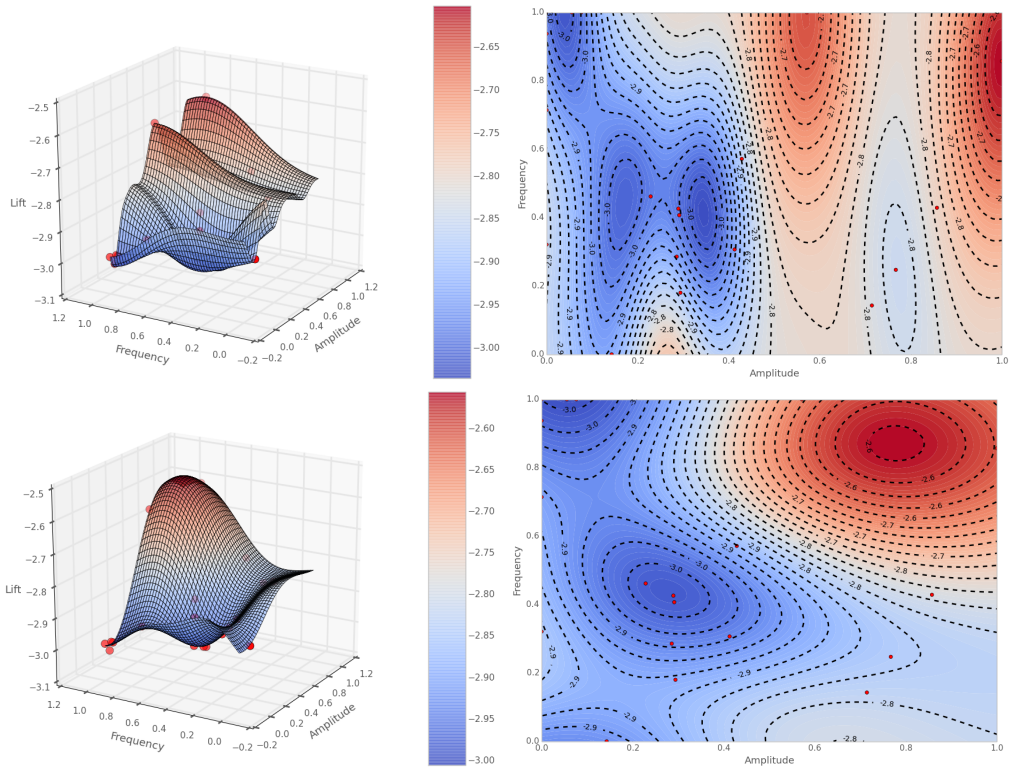


Figure 9.8: Metamodels for the k-ω-SST closure after 11 optimization steps with EI on the NACA0015 configuration. The first figure correspond to an error of $\tau_i^2 = 0$ and the second to $\tau_i^2 = 5 \times 10^{-3}$

9.2.2 EI/PI comparison

From 11 cost function evaluation, the error being identified, the optimization is resumed for both EI and PI with an error estimation variance of $\tau_i^2 = 5 \times 10^{-3}$. The optimization convergence history for EI and PI are shown on Fig. 9.9. For both EI and PI, the optimization is really effective during the first iterations as the cost function is significantly reduced. The Spalart-Allmaras with PI was stopped after 6 iterations due to a lack of evaluation time. The PI merit function tends to reevaluate previously visited points leading to a slowdown of overall convergence. As PI does 3 evaluations per iterations, with respect to the number of solver evaluations, EI performs significantly better than PI.

The optimal points for EI and PI are given on Tab. 9.4. Effective parameters were found as all the optimal points have a better lift coefficient than the baseline configuration ($C_l = 1.34$ for the Spalart-Allmaras closure and $C_l = 1.26$ for the k-ω SST closure).

Because of the computational time requirement, approximately two days per points, the optimization could not be run until complete convergence. Nevertheless, the optimizer with EI has found for the k-ω closure two distinct optima in the search domain. These

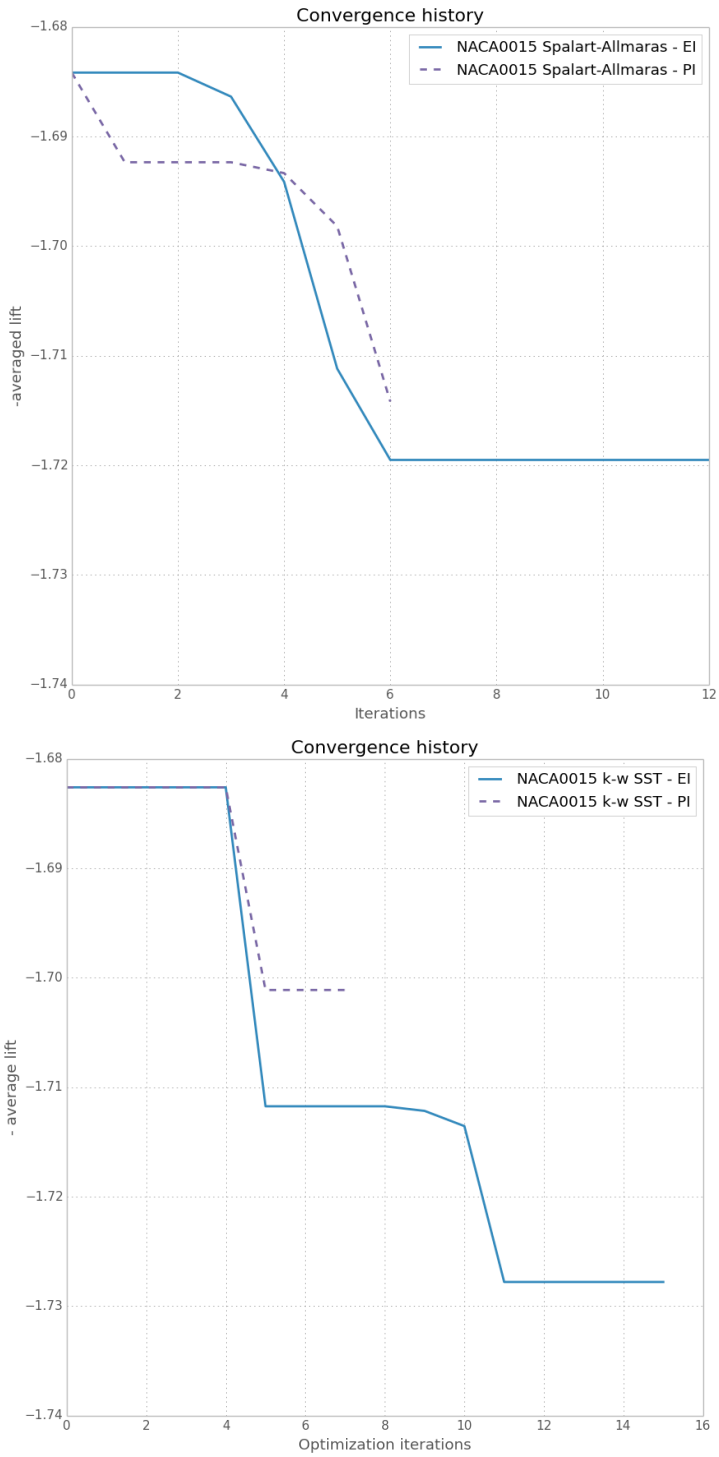


Figure 9.9: Convergence histories for the Spalart-Allmaras and k-ω-SST closures with EI and PI on the NACA0015 configuration.

Turbulence closure	U_{jet}/U_{inf}	$f_{jet}L_{ref}/U_{inf}$	Lift coefficient	Relative gain
Spalart-Allmaras - EI	0.54	1.99	1.72	22%
Spalart-Allmaras - PI	0.57	1.90	1.71	21%
k-ω-SST - EI 1	0.93	1.14	1.71	26%
k-ω-SST - EI 2	0.58	1.99	1.73	27%
k-ω-SST - PI	1.20	1.12	1.70	25%

Table 9.4: Jet parameters, cost function and relative gain with respect to the baseline configuration.

optima are located at mid frequency and mid amplitude, and at high frequency and low amplitude. For the PI merit function, an optimum located at low amplitude and high frequency near one of the EI optima has been found, but with a higher CPU cost. Concerning the Spalart-Allmaras closure, only one optimum could be located at high frequency and low amplitude for both EI and PI merit functions.

Fig. 9.10 and Fig. 9.11 show the meta-models generated using the data from Spalart-Allmaras and k-ω SST closures for both EI and PI. The Spalart-Allmaras meta-models

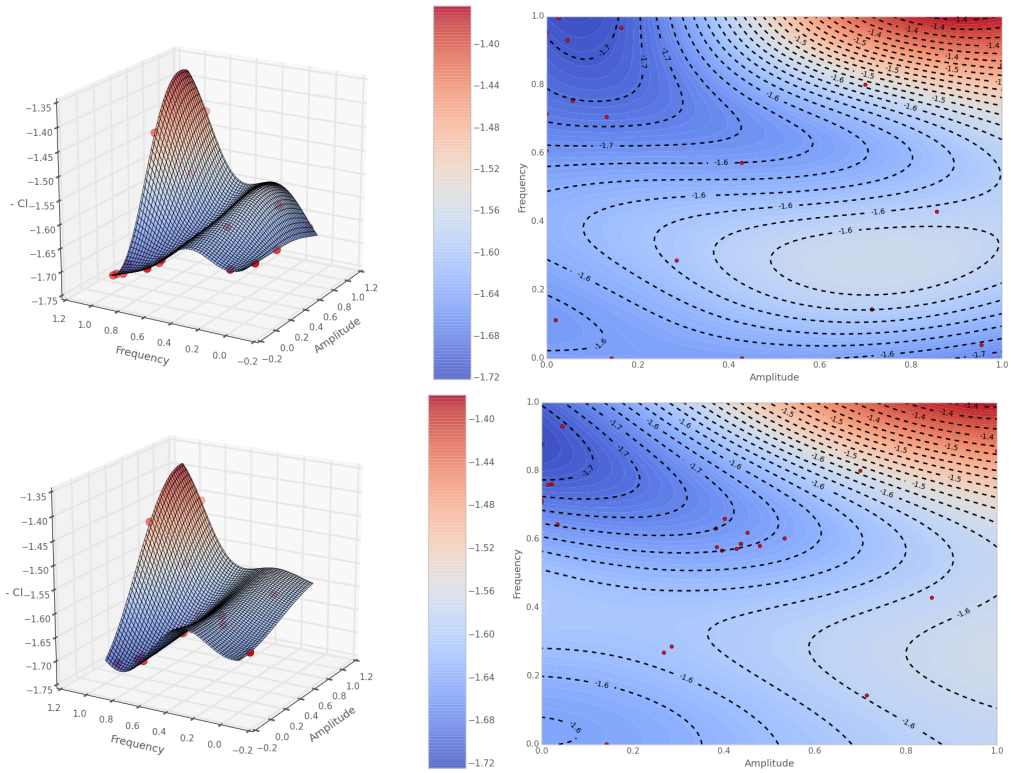


Figure 9.10: Metamodels generated from EGO optimization with EI (top) and PI (bottom) merit functions using the Spalart-Allmaras turbulence closure.

using EI and PI are very similar in shape. They are smooth and present only one

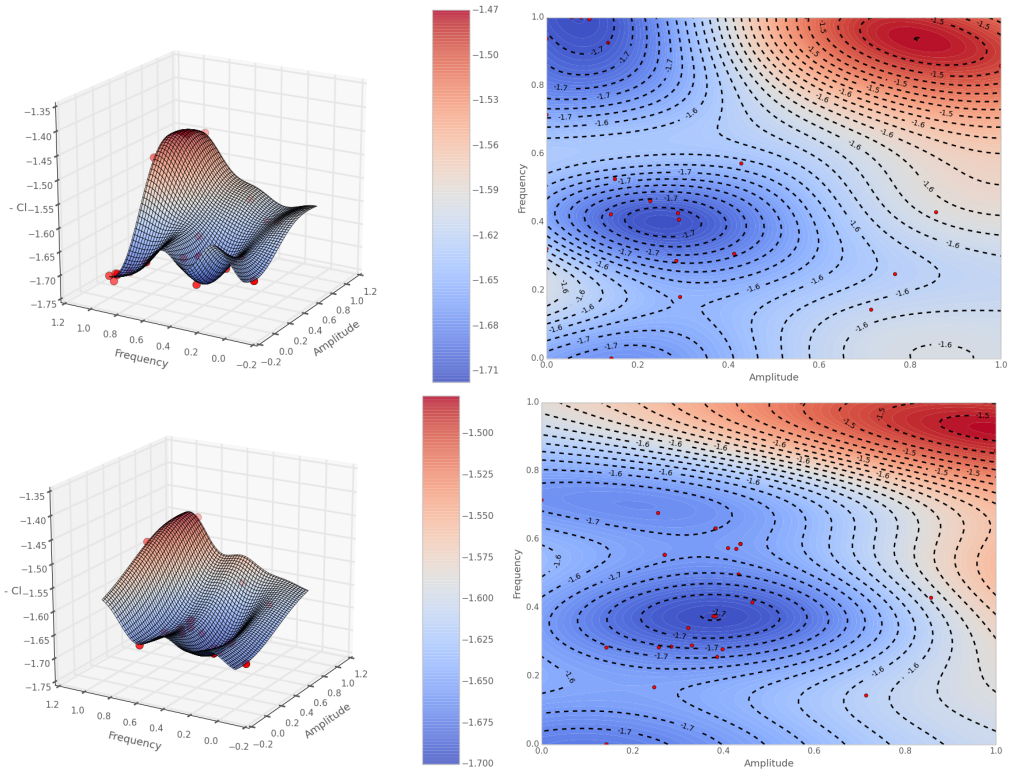


Figure 9.11: Metamodels generated from EGO optimization with EI (top) and PI (bottom) merit functions using the k-ω SST turbulence closure.

optimum, unfortunately, at the border of the search domain. Compared to the k-ω SST turbulence model, the meta-model does not have any optimum located in the mid-range amplitude and frequency. Some differences exist between the EI and PI meta-models in the low amplitude/high frequency area. Indeed, the PI is lacking points within this region making it very uncertain at this range of amplitudes and frequencies. The PI merit function tends to cluster the points around specific areas without enough exploring unknown areas.

It is interesting to note that the turbulence closure seems to have a great impact on the meta-model and on the optima found by the optimizer. Indeed, the Spalart-Allmaras closure does not have any mid-range optimum as for the k-ω SST turbulence closure. The impact of turbulence will be described in more details in the next optimization test case.

Because of the better results the EI merit function, the following is only based on EI results only.

9.2.3 Analysis of optimum controlled flows

Four snapshots of the vorticity and turbulent viscosity or Spalart-Allmaras variable corresponding to the zero blowing phase ($\phi = 0$), maximum blowing phase ($\phi = \pi/2$), zero suction phase ($\phi = \pi$) and maximum suction phase ($\phi = 3\pi/2$) are shown on Fig. 9.12 to Fig. 9.16 for respectively the optimum found using Spalart-Allmaras, k-ω SST at mid-ranges and k-ω SST at low amplitude and high frequency.

On these figures, it can be seen that the mechanism to increase the lift for the Spalart-Allmaras and the second optimal point of k-ω SST turbulence models is the same. Small oscillations are introduced in the flow, leading the birth of small vortices in the separated flow. The resulting flows look similar to Kelvin-Helmholtz instabilities in mixing layers. On the other hand, the first point of the k-ω SST turbulence closure shows bigger vortices leading a more complex flow as based on vortices separation and interaction at the upper surface, near the trailing edge.

The phase-locked pressure distributions for NACA0015 airfoil are shown of Fig. 9.18, Fig. 9.19, and Fig. 9.20 for respectively the EI optimum with Spalart-Allmaras turbulence closure, EI 1 and EI 2 optima with the k-ω SST turbulence closure.

These figures show that the controls with high frequency and low amplitude, for both Spalart-Allmaras and EI 2 optimum with k-ω SST turbulence closures, give a very regular pressure distribution past the synthetic jet. These controls provide a very low perturbed pressure field along the airfoil. On the other side, the EI 1 optimum, with mid-range control parameters using k-ω SST turbulence closure gives higher variation of pressure convected downstream, past the synthetic jet.

Time averaged flow pressure coefficient for the Spalart-Allmaras and k-ω SST closures are shown on Fig. 9.21 and Fig. 9.22. The pressure coefficient past the synthetic jet is lower compared to the baseline configuration. At the leading edge, the suction peak is increased by the presence of the jet. The two synthetic jet parameters gives a very similar pressure coefficient response. The optimum EI 1 has a lower pressure at the trailing edge and a higher pressure at the leading edge than the optimum EI 2. This effect near the trailing edge can be explained by the fact that the bigger vortices generated by EI 2 are not damped as quickly as the perturbations introduced by the control optimum EI 1. For the optimum found using the Spalart-Allmaras turbulence closure, the same conclusions as EI 1 hold.

The time averaged velocity are shown on Fig. 9.23, to Fig. 9.28. Compared to the non controlled case, all the optimum control give a shorter recirculation zone. The detachment point is slightly delayed and the end of the recirculation bubble is more vertical.

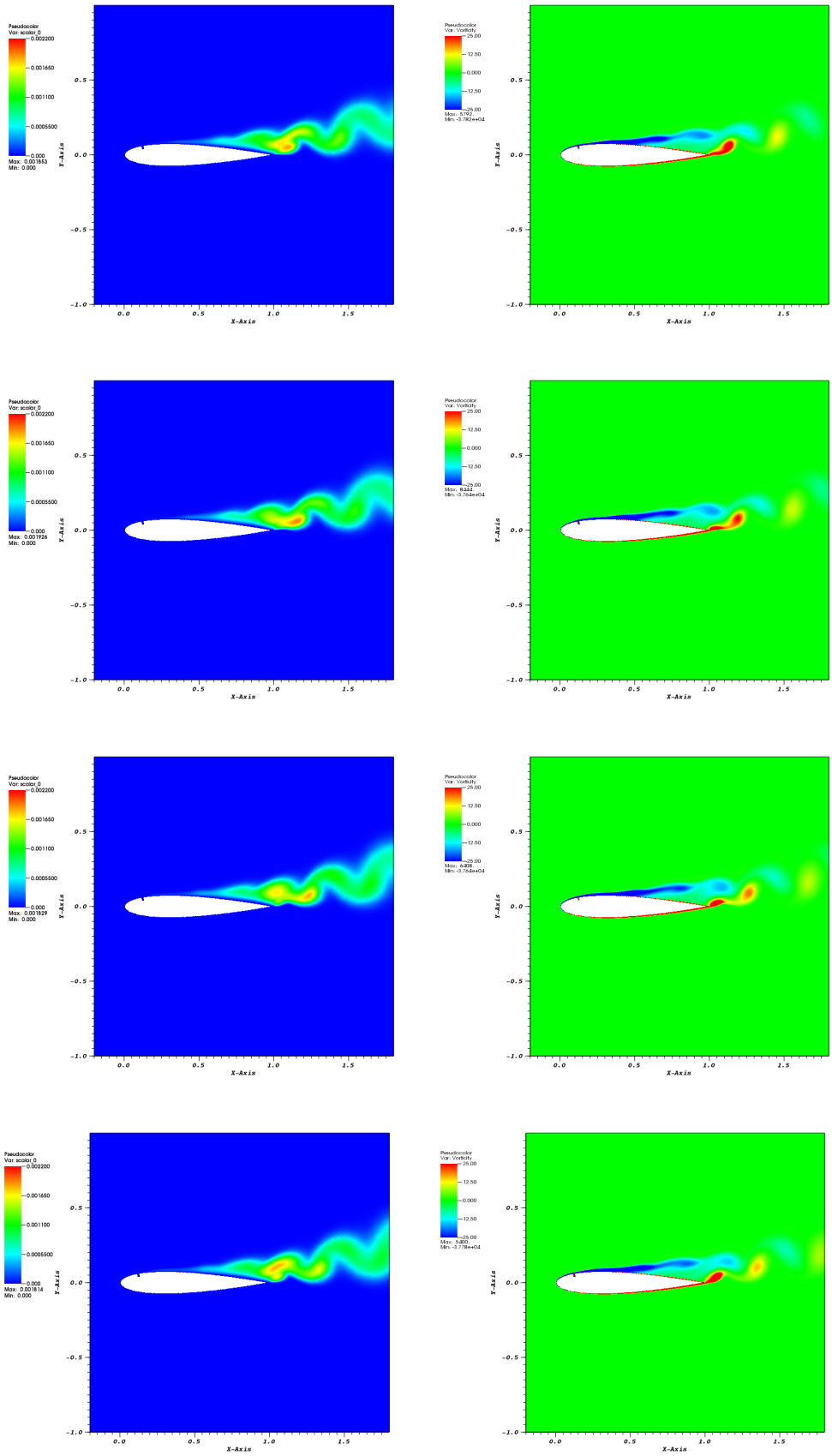


Figure 9.12: Vorticity and turbulent viscosity for the optimum (0.54, 1.99) using Spalart-Allmaras turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

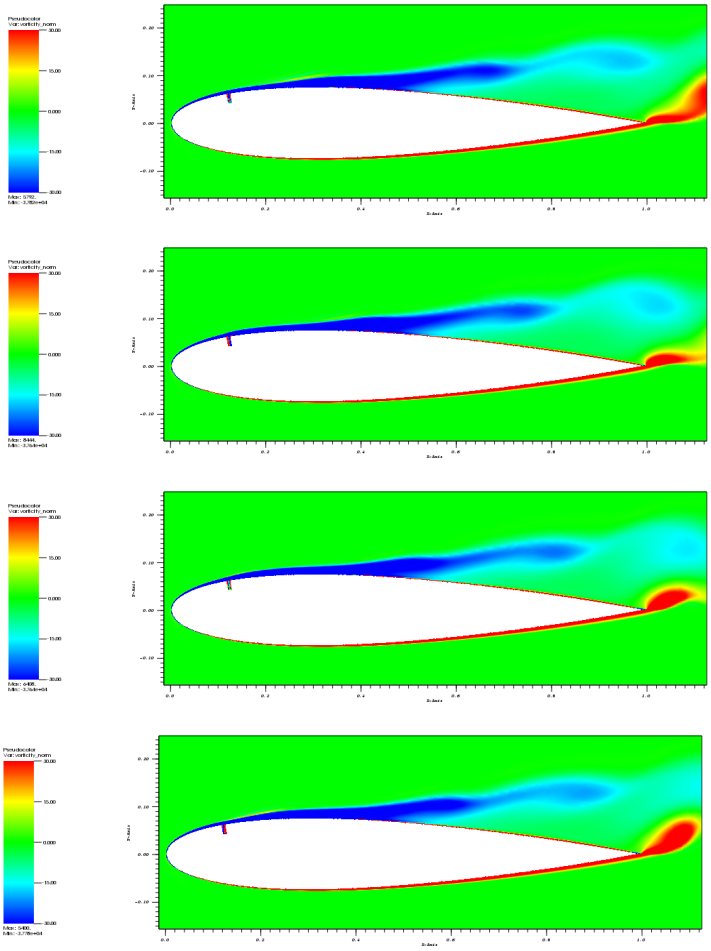


Figure 9.13: Close up vorticity for the optimum (0.54, 1.99) using Spalart-Allmaras turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

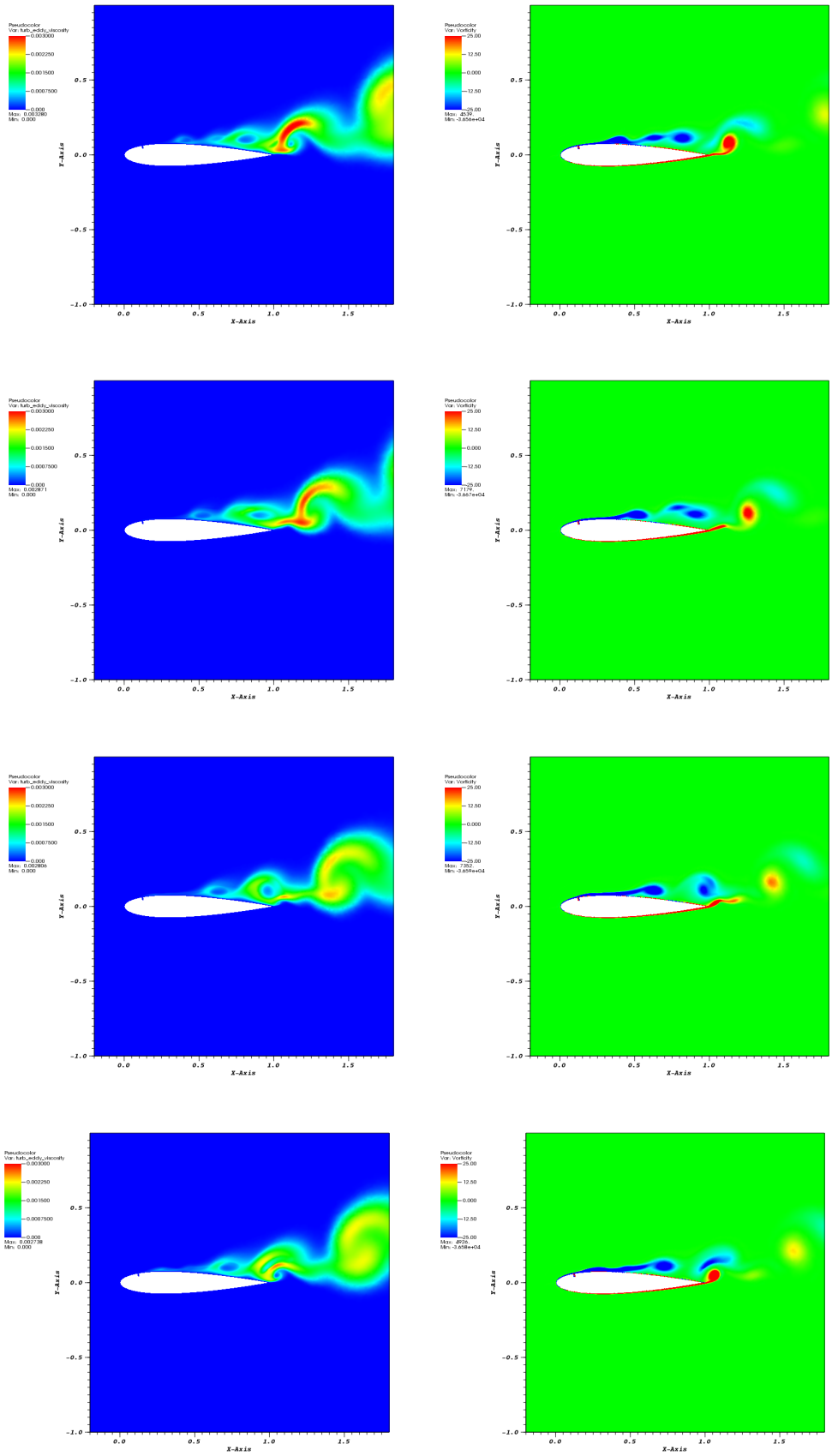


Figure 9.14: Vorticity and turbulent viscosity for the mid-range optimum (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

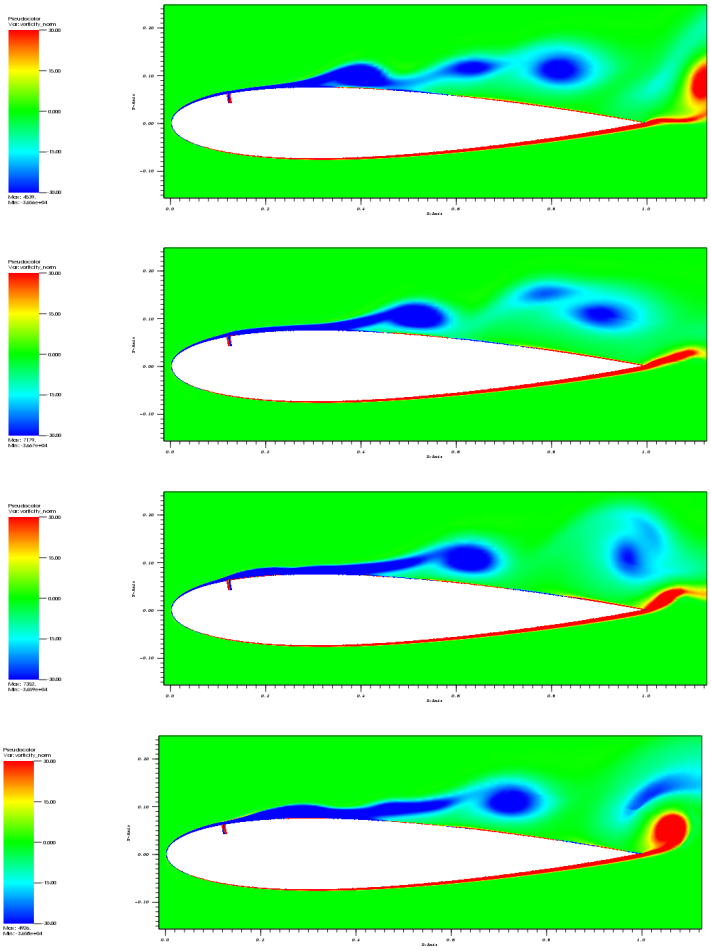


Figure 9.15: Close up vorticity for the mid-range optimum (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

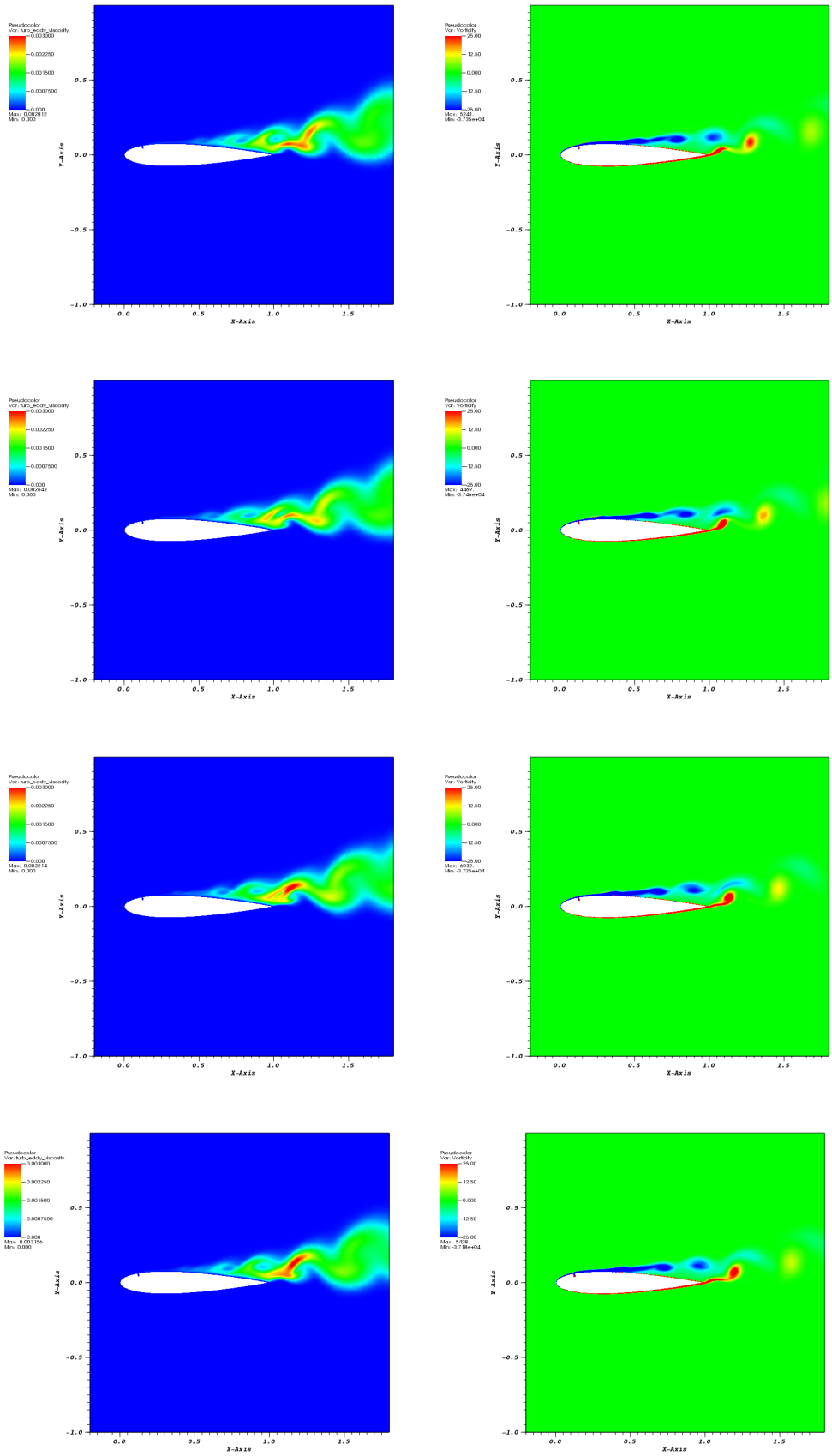


Figure 9.16: Vorticity and turbulent viscosity for the optimum (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

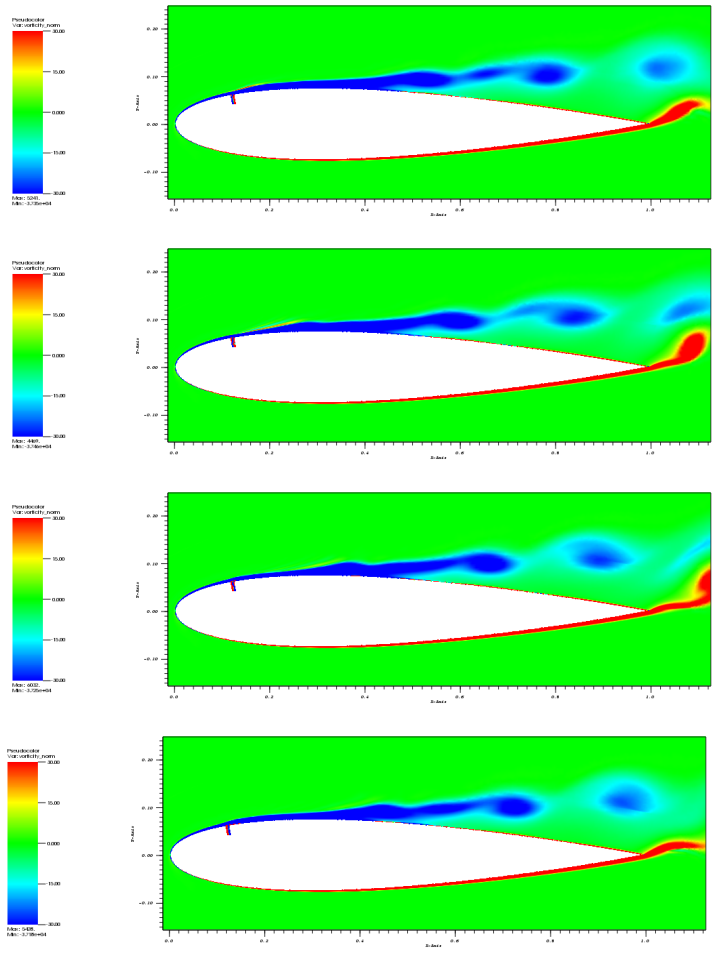


Figure 9.17: Close up vorticity for the (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration. From top to bottom: $\Phi = 0, \pi/2, \pi, 3\pi/2$.

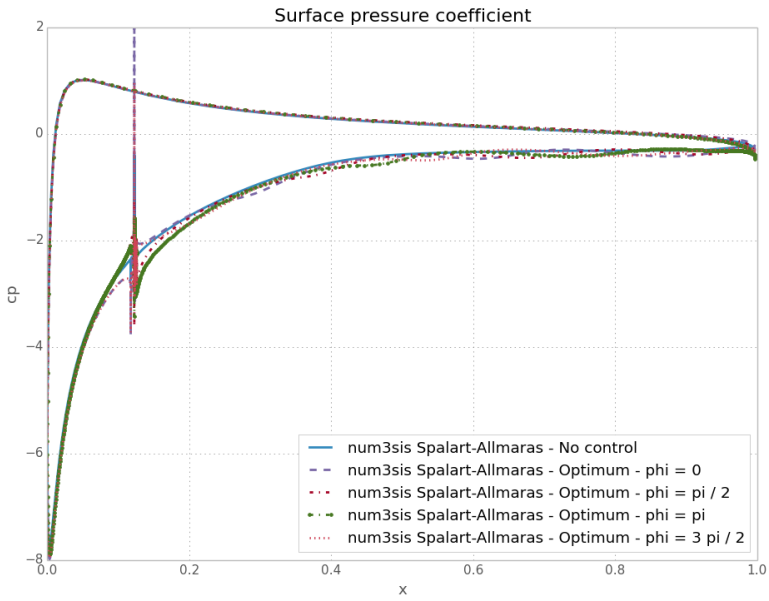


Figure 9.18: Phase-locked pressure distributions at $\Phi = 0, \pi/2, \pi, 3\pi/2$ for the optimum EI 1 (0.54, 1.99) using Spalart-Allmaras turbulence closure on the NACA0015 configuration.

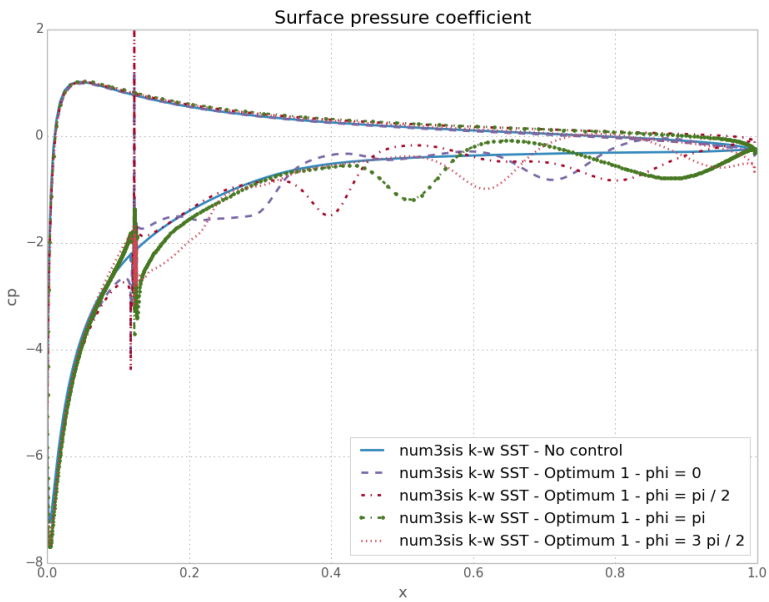


Figure 9.19: Phase-locked pressure distributions at $\Phi = 0, \pi/2, \pi, 3\pi/2$ for the optimum EI 1 (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration.

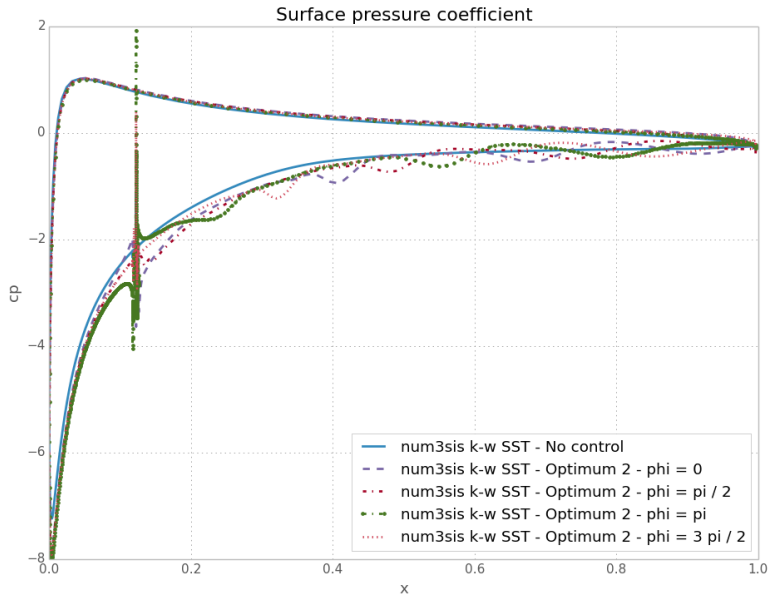


Figure 9.20: Phase-locked pressure distributions at $\Phi = 0, \pi/2, \pi, 3\pi/2$ for the optimum EI 2 (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration.

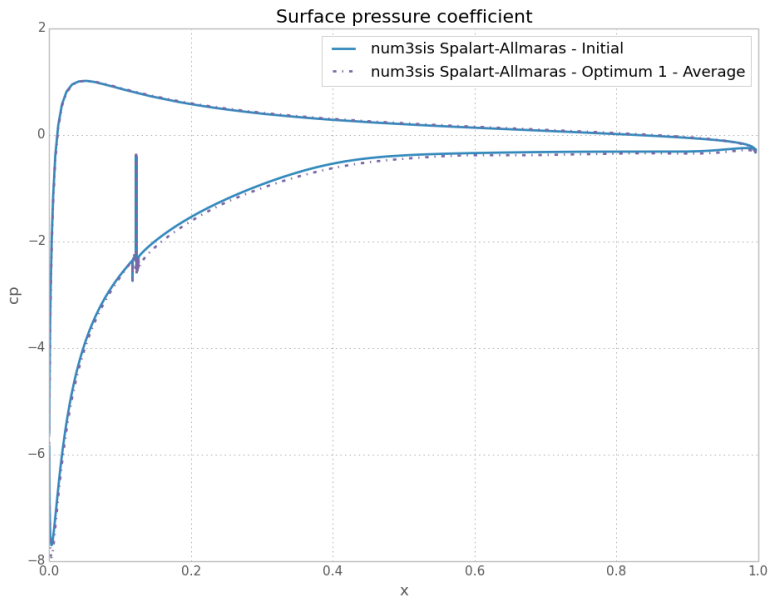


Figure 9.21: Pressure distributions for the time averaged flow of the optima found using EI and Spalart-Allmaras turbulence closure on the NACA0015 configuration.

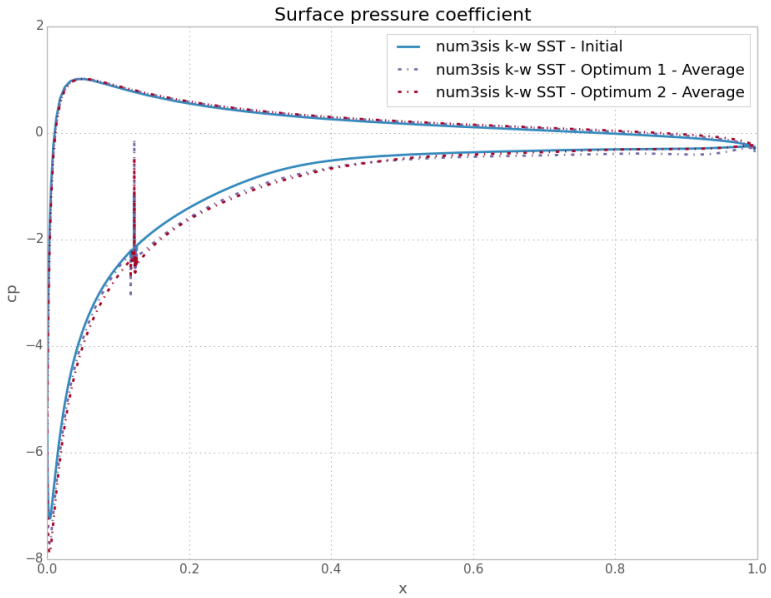


Figure 9.22: Pressure distributions for the time averaged flow of the optima found using EI and k-ω SST turbulence closure on the NACA0015 configuration.

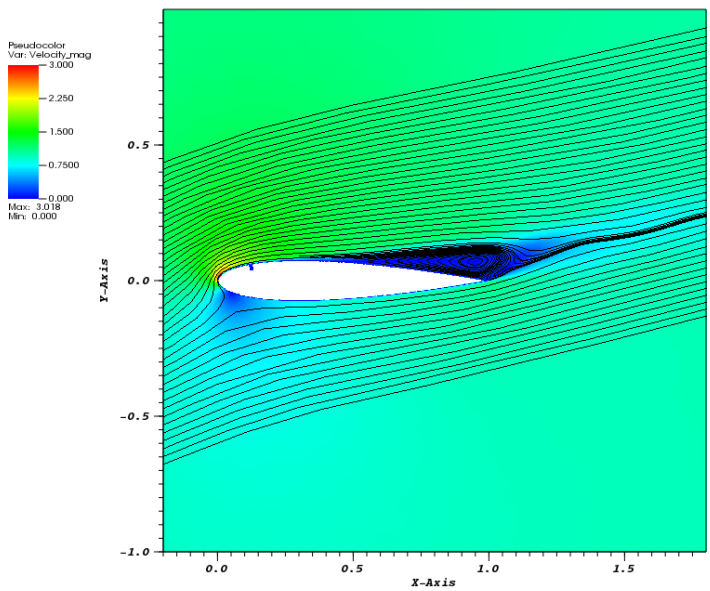


Figure 9.23: Velocity magnitude field for the time averaged flow of the optima found using EI and Spalart-Allmaras turbulence closure on the NACA0015 configuration.

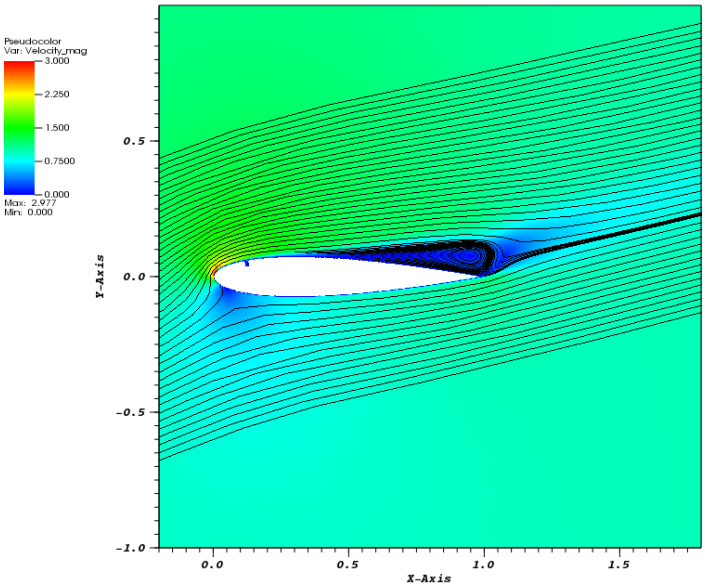


Figure 9.24: Velocity magnitude field for the time averaged flow for the optimum EI 1 (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration.

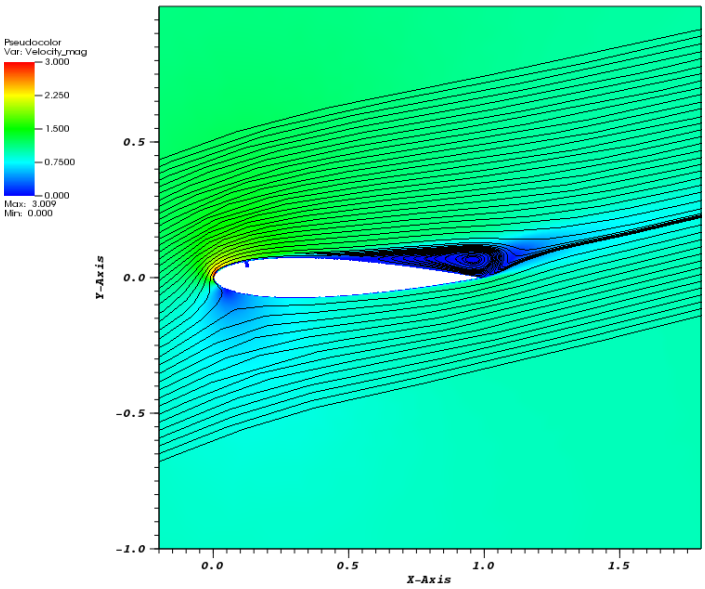


Figure 9.25: Velocity magnitude field for the time averaged flow for the optimum EI 2 (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration.

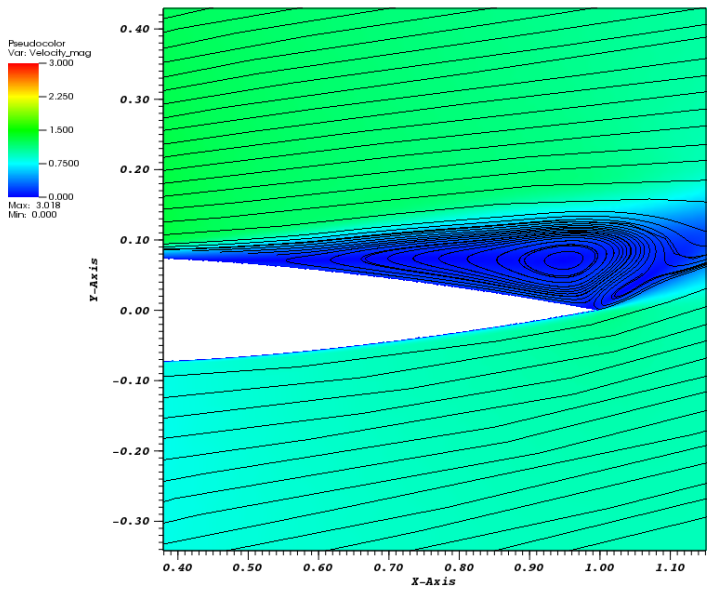


Figure 9.26: Recirculation zoom for the velocity magnitude field for the time averaged flow of the optima found using EI and Spalart-Allmaras turbulence closure on the NACA0015 configuration.

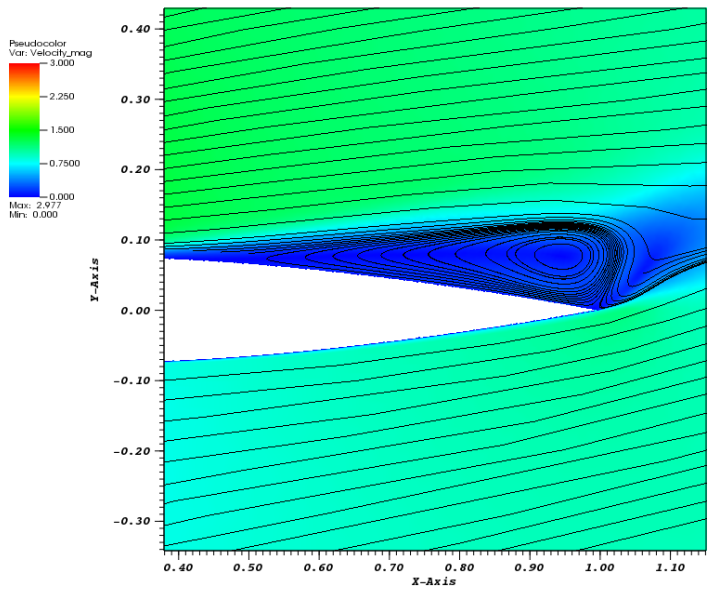


Figure 9.27: Recirculation zoom for the velocity magnitude field for the time averaged flow for the optimum EI 1 (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration.

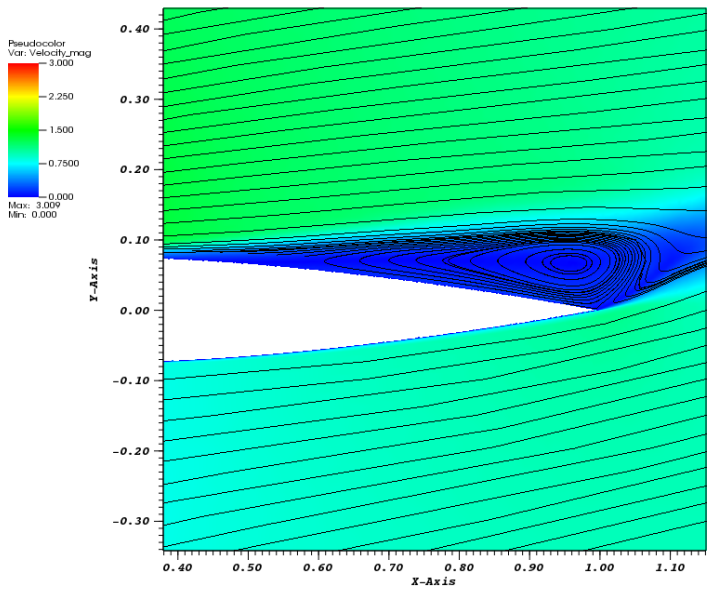


Figure 9.28: Recirculation zoom for the velocity magnitude field for the time averaged flow for the optimum EI 2 (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration.

The Spalart-Allmaras variable and the turbulent kinematic energy for the k-ω SST turbulence closure are shown on Fig. 9.29 to Fig. 9.33 for the non controlled and controlled with optimum configurations.

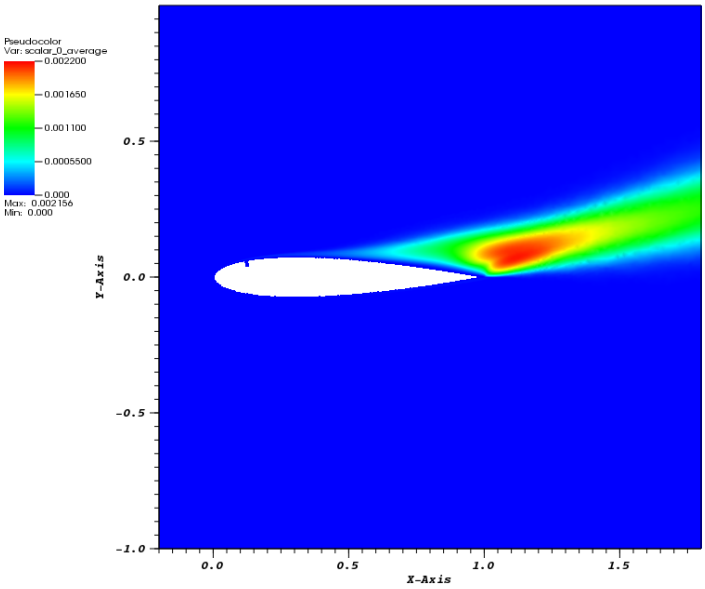


Figure 9.29: Spalart-Allmaras variable field of the non controlled Spalart-Allmaras turbulence closure on the NACA0015 configuration.

The Spalart-Allmaras variable for the Spalart-Allmaras turbulence closure, using the optimum controlled parameters looks very similar above the airfoil compared to the non controlled configuration. Furthermore, this quantity is largely reduced just after the trailing edge. The turbulent kinetic energy fields for the k-ω SST turbulence closure of the controlled optima configurations are weaker than for the non-controlled configuration. The type of the control has a great impact on its distribution. As seen here, in general, the optima controlled configurations have a lower turbulent eddy viscosity than in the initial configurations.

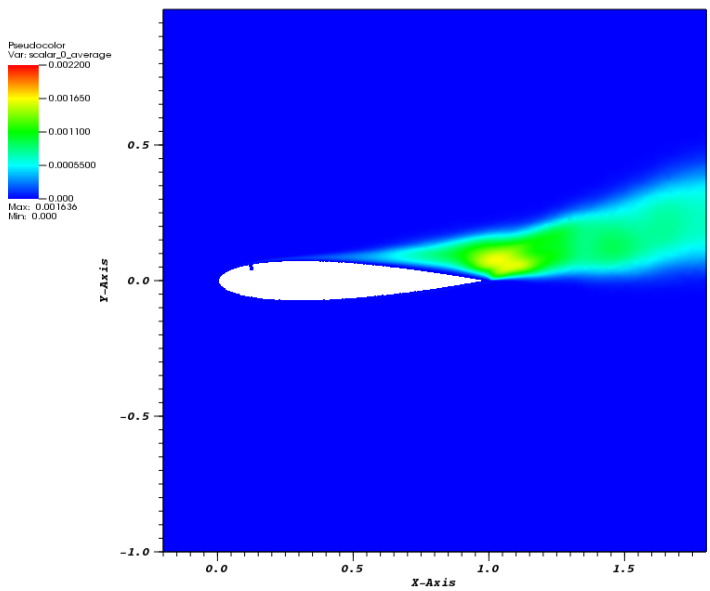


Figure 9.30: Spalart-Allmaras variable field for the time averaged flow of the optima found using EI and Spalart-Allmaras turbulence closure on the NACA0015 configuration.

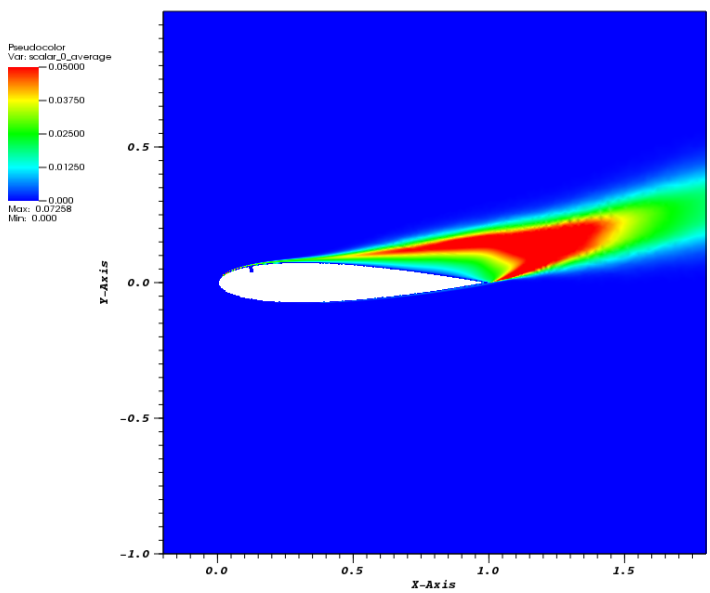


Figure 9.31: Turbulent kinetic energy field of the non controlled turbulent kinetic energy using k-ω SST turbulence closure on the NACA0015 configuration.

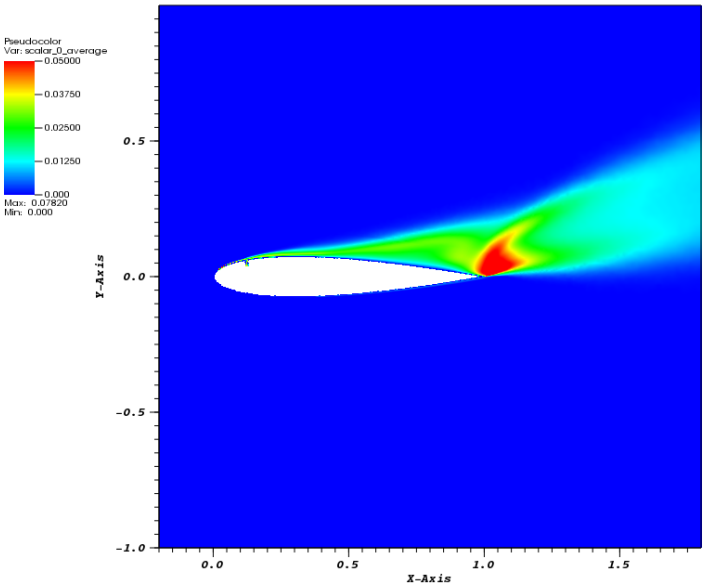


Figure 9.32: Turbulent kinetic energy field for the time averaged flow for the optimum EI 1 (0.93, 1.14) using k-ω SST turbulence closure on the NACA0015 configuration.

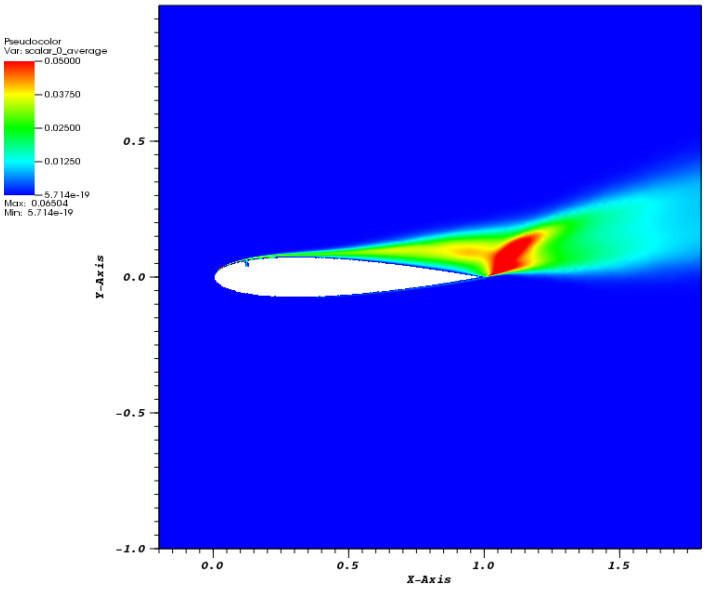


Figure 9.33: Turbulent kinetic energy field for the time averaged flow for the optimum EI 2 (0.58, 1.99) using k-ω SST turbulence closure on the NACA0015 configuration.

9.3 Discussion

In this study, a global optimization of amplitude and frequency of a synthetic jet have been performed. It has been shown that taking into account cost function observations noise is mandatory to retrieve a correct meta-model. After having chosen an appropriate simulation error, EI and PI merit functions have been applied on the optimization problem with success. Accounting of the limited number of evaluations, using k-ω SST turbulence closure, the EI merit function showed better results than PI as it was able to localize two parameters optima while PI only found one optimum without exploring other interesting part of the search domain.

An increase in lift of approximately 20% could be achieved for both the Spalart-Allmaras and k-ω SST turbulence closures. Such order of magnitude in lift coefficient improvements using synthetic jets are experimentally [Seifert 1999] and numerically [Donovan 1998] observed for post stall configurations, although in the given references, synthetic jets for the NACA0015 case are placed at the leading edge and not on the upper surface as in our experimentations. In these papers, the frequency used to control the lift provides well detached vortices that are convected downstream the airfoil very similar to the first optimal point of the k-ω SST turbulence found using EI. Such frequencies tends to excite the natural shedding frequency (not observed here at 18 degrees, using URANS simulations). The natural shedding frequency for a $\alpha = 22$ deg NACA0015 configuration at comparable Reynolds number is reported to be $f_{shed}L_{ref}/U_{inf} \approx 0.53$, and an optimal actuation frequency $f_{jet}L_{TE}/U_{inf} \approx 1$ with L_{TE} the distance between the trailing edge and the actuation. Even if the case considered here is not exactly the same, the first optimum found by EI ($f_{jet}L_{ref}/U_{inf} = 1.14$) gives $f_{jet}L_{TE}/U_{inf} = 1.003$, being the reported as an effective frequency close to unity value by Seifert [Seifert 1996]. Duvigneau and al. in [Duvigneau 2006] reports optimal frequency of $f_{jet}L_{ref}/U_{inf} = 0.85$ and an amplitude of $U_{jet}/U_{inf} = 1.72$ for the optimization of a synthetic jet over a NACA0015 airfoil at 18 degrees using k-ω SST closure. These parameters optima can not be seen on the present meta-models. This discrepancy between the present work and the reported optima values might occur because the jet angle was also optimized and attained a value of 30 degrees. In [Seifert 1999], it also has been reported that lower forcing levels are only effective with actuation frequency of $f_{jet}L_{TE}/U_{inf} > 2$. This result could also be noticed in our optimization experiments, using the Spalart-Allmaras closure and the k-ω SST closures (EI 2), where the optima control parameters found are $f_{jet}L_{ref}/U_{inf} \approx 2$ and $U_{jet}/U_{inf} \approx 0.5$.

The mechanism to increase lift, is the same for the low amplitude and high frequencies by introducing small perturbation in the flow for both Spalart-Allmaras and k-ω SST closures. On the other side, for the medium range control parameters, the k-ω SST closures provides an optimum where bigger vortices are generated and convected downstream, leading to a different mechanism.

The meta-models differ significantly in the frequency and amplitude middle ranges for the two turbulence closures, leading to an absent optimum using the Spalart-Allmaras

meta-model. According to these results, the impact of turbulence closure can have a big influence on the optimization process. For this reason, a particular study of the impact of turbulence closure is carried out on the Backward Facing Step case and presented in the next chapter.

Chapter 10

Backward Facing Step

Contents

10.1 Incompressible turbulent Navier-Stokes equations 155

10.2 Backward Facing Step 156

10.2.1 Non controlled case 158

10.2.2 Controlled case validation 163

10.3 A naive optimization neglecting observation error 170

10.4 Impact of observation variance 171

10.5 Impact of turbulence closures 173

10.5.1 Optimization results 173

10.5.2 Cross-validation 177

10.6 Discussion 178

A second test case is considered for which a detached flow imposed by the geometry on a simple configuration. This test case is used to further study the influence of turbulence and numerical errors. The study is carried out with the ISIS-CFD solver as a collaboration on control with the Fluid Mechanics Laboratory, CNRS UMR 6598 / ECN.

10.1 Incompressible turbulent Navier-Stokes equations

The Backward Facing Step flow studies are carried out with the ISIS-CFD solver, developed by the CFD group of the Fluid Mechanics Laboratory and available as a part of the FINE^{TM}/Marine computing suite. It solves incompressible Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations. Because of it well validated implementation of turbulence models, it is employed to carry out a precise study of turbulence influence on simulation and optimization of separated flows.

The solver is based on the finite-volume method to build the spatial discretization and solves the equations with a face-based cell-centered approach.

Within this framework, the incompressible conservation laws under isothermal conditions are written as

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_S \rho \boldsymbol{U}\cdot\boldsymbol{n}dS = 0, \tag{10.1}$$

$$\frac{\partial}{\partial t}\int_V \rho U_i dV + \int_S \rho U_i \boldsymbol{U}\cdot\boldsymbol{n}dS = \int_S (\tau_{ij}I_j - pI_i)\cdot\boldsymbol{n}dS, \tag{10.2}$$

where V is the domain of interest or control volume, bounded by the closed surface S with a unit normal vector $\boldsymbol{n}$ directed outward. $\boldsymbol{U}$ and p represent, respectively, the velocity and pressure fields. τ_{ij} are the components of the viscous stress tensor, whereas I_j is a vector whose components are zero, except for the jth component which is equal to unity.

All flow variables are stored at geometric centers of the arbitrary shaped cells. Surface and volume integrals are evaluated according to second-order accurate approximations by using the values of integrand that prevail at the center of the face f, or cell C , and neighbor cells C_{nb}. The various fluxes appearing in the discretized equations (10.1) and (10.2) are built using the Gamma Differencing Scheme (GDS) [Jasak 1996, Queutey 2007]. Other flux determination methods as centered, upwind or hybrid schemes are implemented in ISIS-CFD and more details are detailed by Duvigneau and Visonneau [Duvigneau 2003] and more recently by Queutey and Visonneau [Queutey 2007]. A pressure equation is obtained in the spirit of the Rhie and Chow SIMPLE algorithm [Rhie 1982]. As for the compressible solver, unsteady terms are solved using a dual time stepping approach.

In the case of turbulent flows, additional transport equations for modeled variables are solved in a similar form to the momentum equations and are discretized and solved using the same principles.

10.2 Backward Facing Step

The test-case proposed here corresponds to a classical case of detached flow imposed by the geometry: we consider the two-dimensional flow over a backwards facing step. This test-case is selected here to provide detached flows with a simple geometrical configuration. The configuration conditions described in the experiments carried out by Driver and Seegmiller [Driver 1985] are used in this study. This test case has been documented using incompressible [Le 1997, Yoder 1999] as well as compressible [Rumsey 2008] flows. The objective of the actuation is the reduction of the separation length.

The backward-facing step geometry including a control device is represented on Fig. 10.1. The geometry and parameters set according to Driver case: $h = 0.0127$m, $U_{ref} = 44.2$m/s, $M_{ref} = 0.128$, the Reynolds number based on the boundary layer momentum thickness prior to the step is set to $Re_\theta = 5000$ and the boundary layer thickness at the inlet is $\delta_{BL} = 0.019$m. The control device is a suction / blowing jet with a diameter of $h/10$ and is located at $h/50$ from the step.

As boundary conditions, a turbulent velocity profile corresponding to experiments is imposed at inlet, and a uniform pressure condition at outlet. The jet boundary condition is imposed using a boundary condition model. The parameters of the actuation are the

Jet parameters	Min	Max
Amplitude (m/s)	4	50
Frequency (Hz)	50	1000

Table 10.1: Jet parameters for the optimization over the Backward Facing Step.

amplitude U_{jet} and the frequency f_{jet}. They are allowed to vary in the following intervals: 4m/s $\leq U_{jet} \leq$ 50m/s and 50Hz $\leq f_{jet} \leq$ 1000Hz.

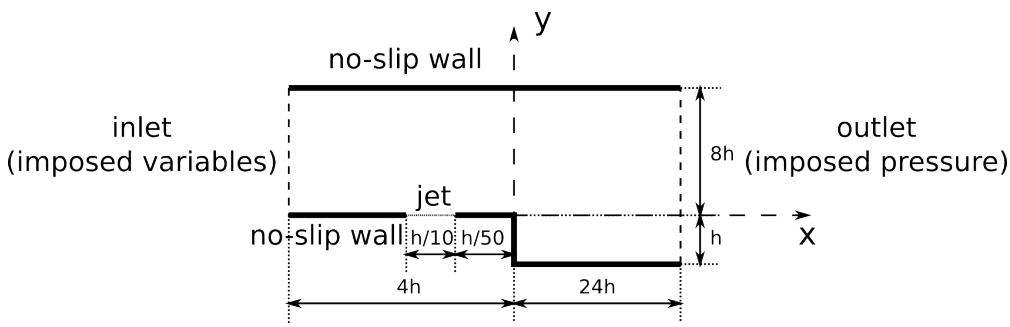


Figure 10.1: Backward-facing step with actuation configuration.

Three different meshes have been designed for this particular configuration to assess grid convergence. In the incompressible case, the coarse, medium and fine mesh respectively contains 29778, 45805 and 77064 nodes. In all cases, the boundary layer is refined at the wall to fulfill the $y^+ < 1$ criterion at all time steps, at all locations and for all closures. All grids include a refined area between the step and the location $9h$, as illustrated in Fig. 10.2. An external tool is used to fulfill the parameters of the test case. For inlet, all the primitive variables are imposed from the boundary profile. For the outlet, the pressure only is imposed and other primitive variables are extrapolated. No-slip wall boundary condition are used for all walls.

As well, three different time steps based on the jet frequency are selected. The large, medium and small time steps are defined so that one period contains respectively 64, 128 and 256 time steps.

The actuator has a width of value $h/10$ and is introduced at a location very close to the step corner (h/50), generating a vertical suction / blowing actuation

The objective function for this optimization test-case is the recirculation length of the time-averaged flow. A post treatment locates where the wall shear stress vanishes using a linear interpolation at the points located at the wall. As design parameters for the optimization, the jet amplitude and frequency are allowed to vary in the intervals described in the Tab. 10.1.

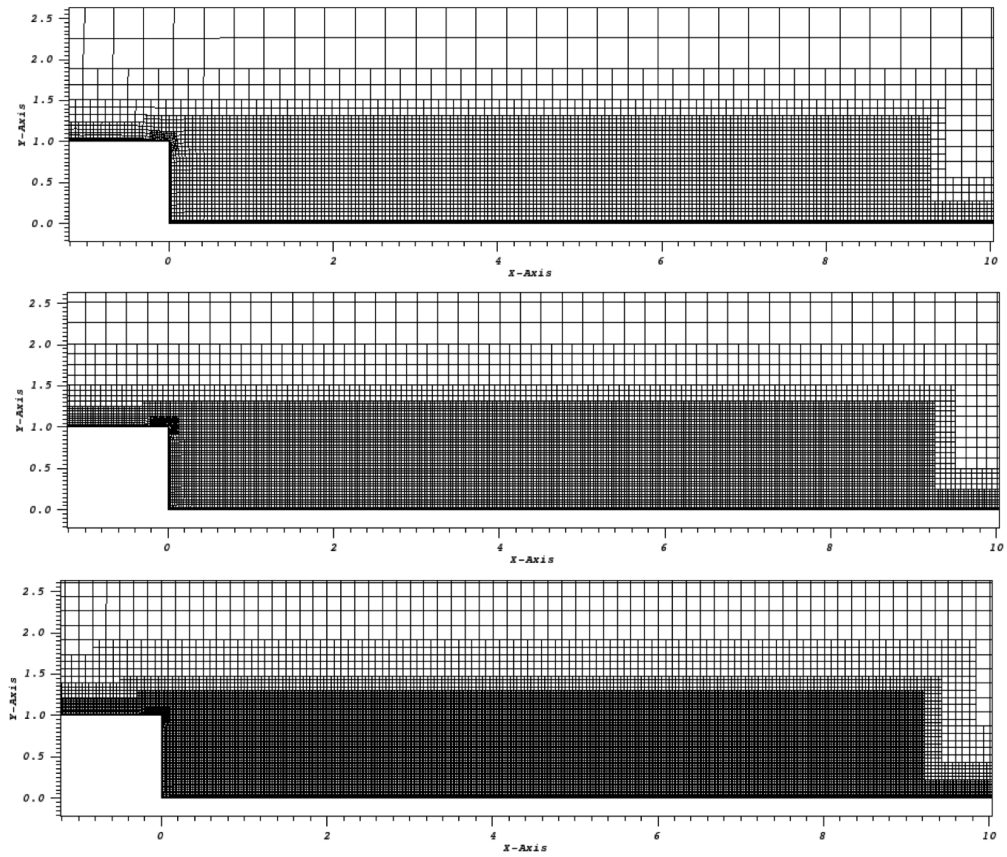


Figure 10.2: Backward-facing step with actuation configuration.

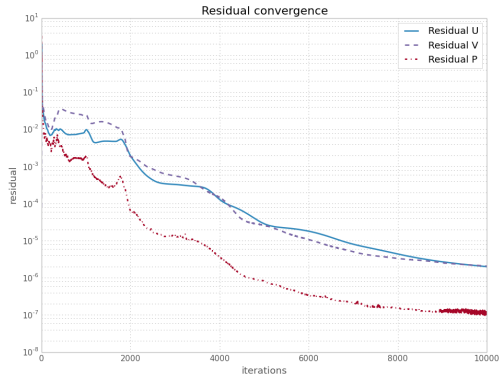
10.2.1 Non controlled case

The characteristics of the flow without actuation are compared for the different turbulence closures considered. In this context, the residuals of the steady solutions on the fine grid are decreased by at least 5 orders as shown on Fig. 10.3. A grid convergence study is carried out for all closures, as illustrated in Fig. 10.4, in terms of separation length. As seen, the discretization error is far lower than the modeling error.

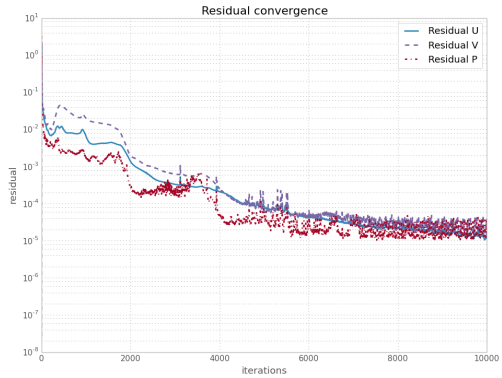
Velocity streamlines different turbulence closures are depicted on Fig. 10.5, velocity profiles on Fig. 10.6, turbulent kinematic eddy viscosity on Fig 10.7 for the finest grid.

The pressure coefficient for each turbulent models is shown on Fig. 10.8.

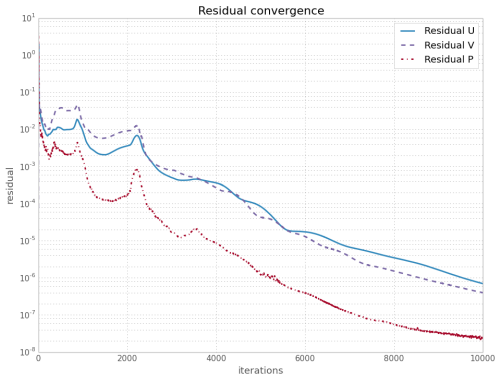
As well, the table 10.2 gives a comparison of the recirculation length for the different models. Although it deals with a simple test-case, without actuation, turbulence closure plays already a critical role in the flow prediction. In particular, k-ω SST Menter is the closest one to the experimental recirculation length, EASM closure overestimates the separation length, while Spalart-Allmaras and Launder-Sharma k-ε ones predict a



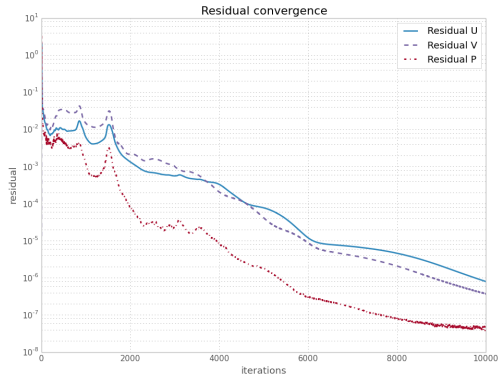
(a) Spalart-Allmaras closure.



(b) k-ε Launder-Sharma closure.



(c) k-ω-SST Menter closure.



(d) k-ω-ne-easm closure.

Figure 10.3: Residuals convergence for stationary computations.

too small recirculation region, with respect to experimental measurements. The Spalart-Allmaras closure has the worst behavior concerning the velocity profiles.

In the next sections, we quantify the effects of the actuation and observe if some common trends can be established, despite of this initial discrepancy.

Experimental	Spalart-Allmaras	k-ε Launder-Sharma	k-ω SST Menter	k-ω ne-easm
6.26	6.05	5.41	6.37	7.56

Table 10.2: Comparison of recirculation length l/h computed on the fine grid against the experimental results from Driver and Seegmiller.

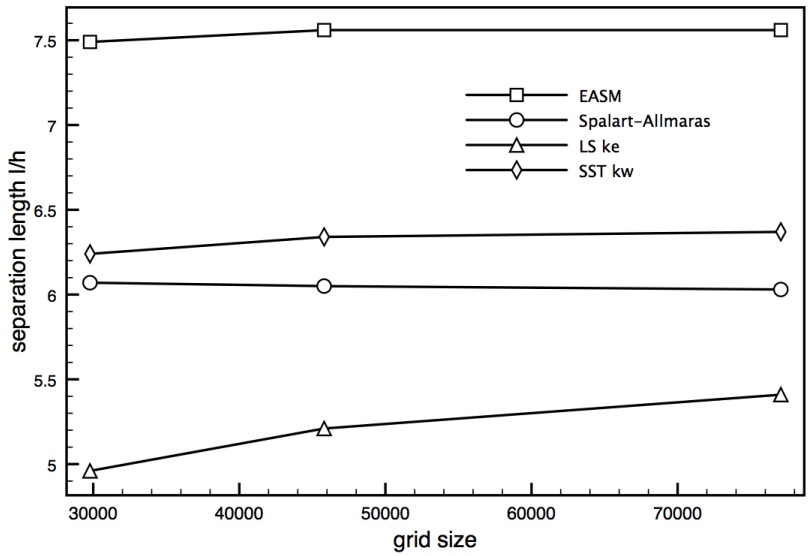
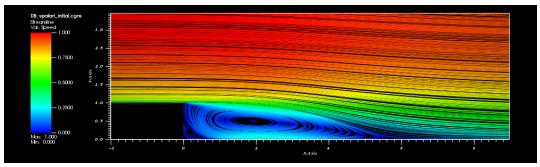
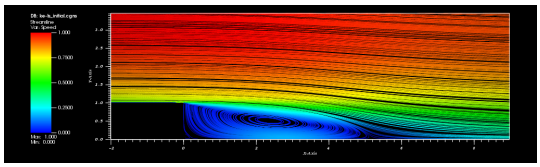


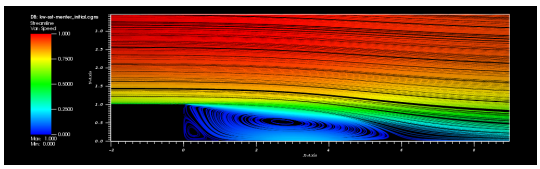
Figure 10.4: Grid convergence study without actuation.



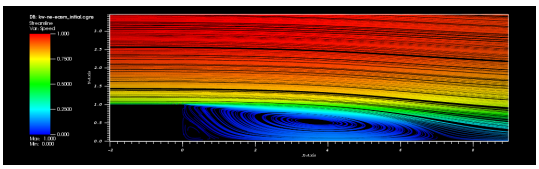
(a) Spalart-Allmaras closure.



(b) k-ε Launder-Sharma closure.



(c) k-ω-SST Menter closure.



(d) k-ω-ne-easm closure.

Figure 10.5: Comparison of velocity streamlines for different turbulence closures for the flow over the backward facing step without control. Simulation done on the fine grid.

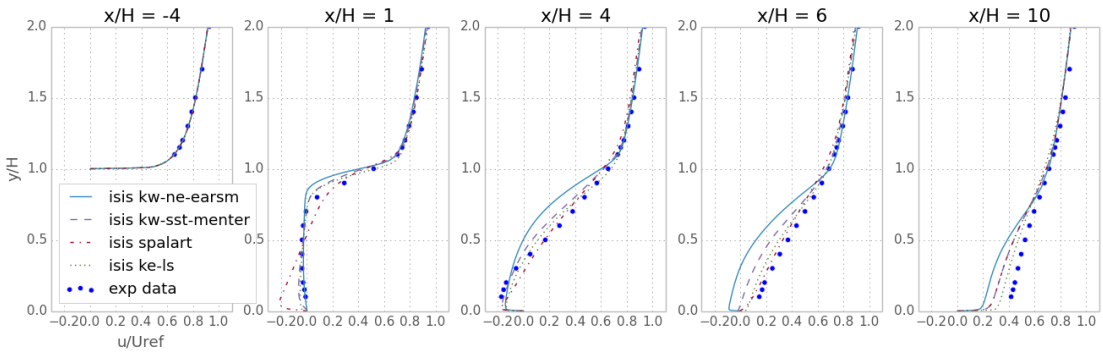
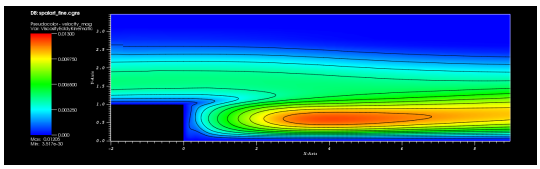
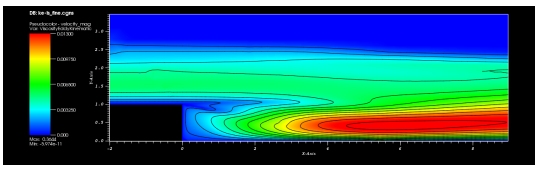


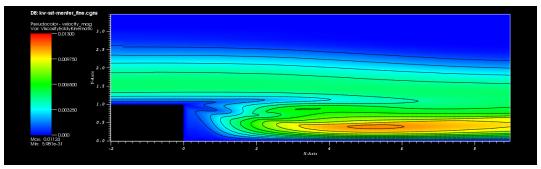
Figure 10.6: Initial velocity profiles at $x = -4$, $x = 1$, $x = 4$, $x = 6$ and $x = 10$ for different turbulence closures for the flow over the backward facing step without control. Simulation done on the fine grid.



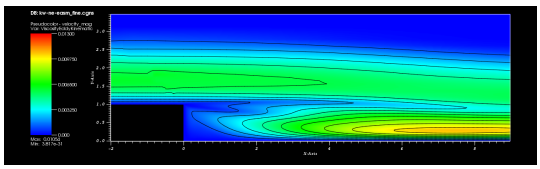
(a) Spalart-Allmaras closure.



(b) k-ε Launder-Sharma closure.



(c) k-ω-SST Menter closure.



(d) k-ω-ne-easm closure.

Figure 10.7: Comparison of kinematic eddy viscosity for different turbulence closures for the flow over the backward facing step without control. Simulation done on the fine grid.

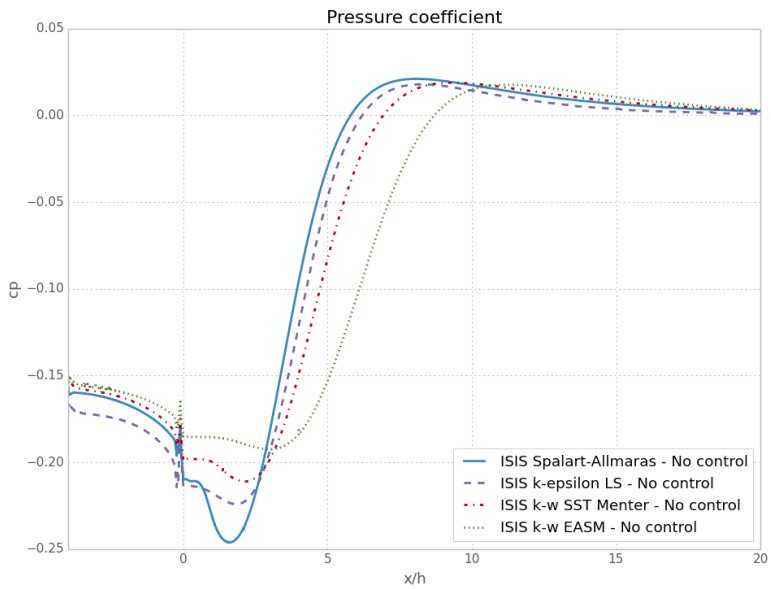


Figure 10.8: Initial pressure coefficient for different turbulence closures of the flow over the backward facing step without control. Simulation done on the fine grid.

10.2.2 Controlled case validation

10.2.2.1 Unsteady convergence validation

Each simulation of actuated flow is performed starting from the steady state flow found without actuation. Unsteady computations are carried out until transient effects have vanished, which is approximately 0.025s as seen on Fig. 10.9.

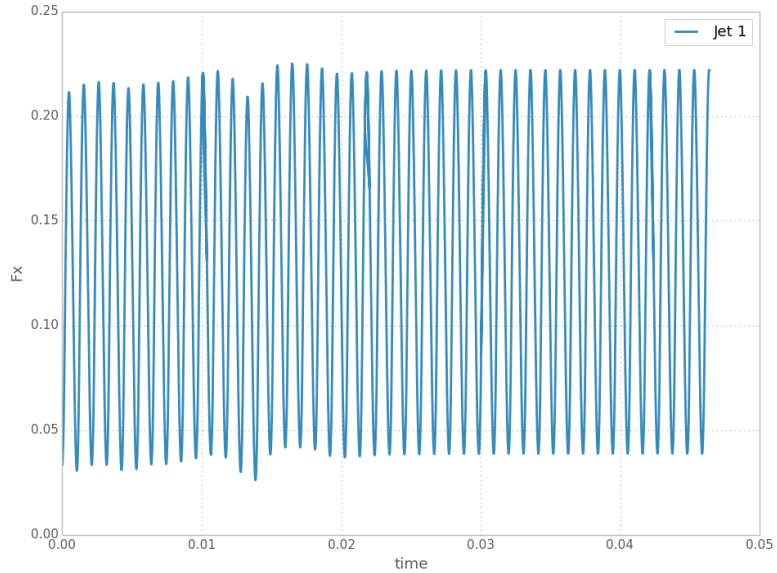


Figure 10.9: Example of the effort Fx response for the simulation of a controlled flow. The transient effects are vanished after 0.25s.

The time step employed is scaled on the actuation frequency, so that all configurations count the same number of time steps per actuation period. For each time step, a reduction of three orders of non-linear residuals is used as stopping criterion. The separation length is computed afterwards, from the time-averaged flow, on the basis of a four jet period integration starting after 0.035s. The use of the time averaged flow is justified by the fact that the location of the reattachment point is a time dependent discontinuous quantity. Furthermore, even with time averaged flow, the regularity of the location of the reattachment point with respect to control parameters is not obvious, as periodic detachment of vortices can appear. The separation length is estimated by locating the point where the skin friction vanishes as seen on Fig. 10.10.

A convergence study, in terms of grid size, time step and dual time stepping order of convergence, is first performed to assess the computations validity. Since the optimization criterion is the separation length, this quantity is used for the comparisons. In the perspective of optimization, the grid, the time step and the dual time stepping order of convergence should be selected in such a way that the variations of the separation length, due to a change of actuation parameters, are accurately quantified. Therefore, a set of

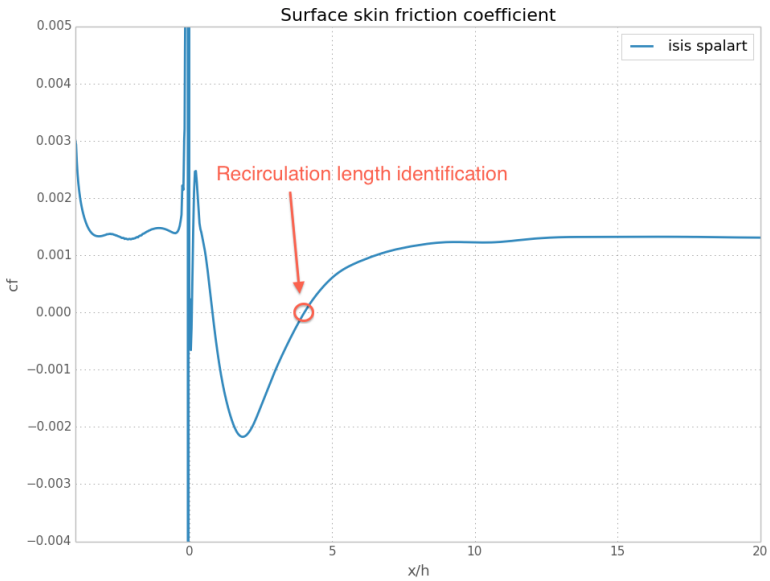


Figure 10.10: Example of an averaged skin friction coefficient. It vanishes at the reattachment point.

actuation configurations are compared.

This study is conducted for the EASM closure only, which is expected to be the most demanding one.

To measure the effects of the grid size on the meta-model, three grids are considered, with a fixed number of 256 time steps per period and a reduction of 4 orders of non linear residuals. Fig. 10.11 presents the evolution of the separation length as the grid size increases. The different meta-models obtained using the three grids can be compared on Fig. 10.12.

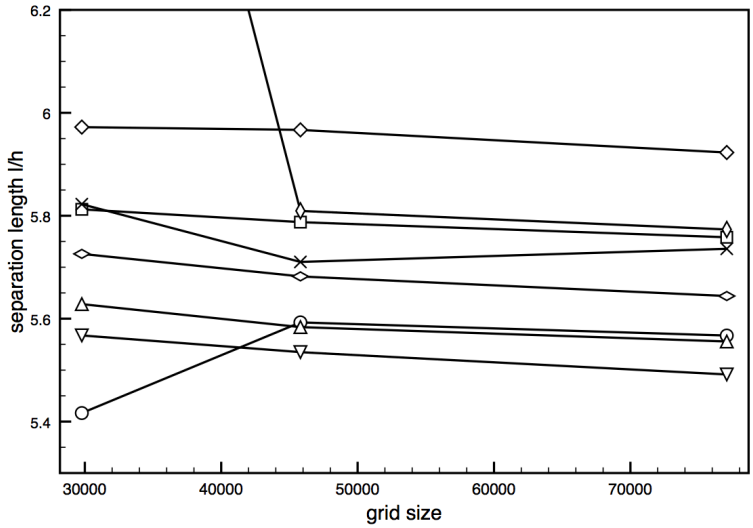
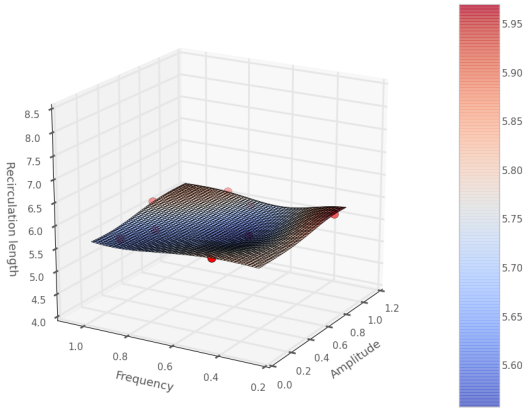
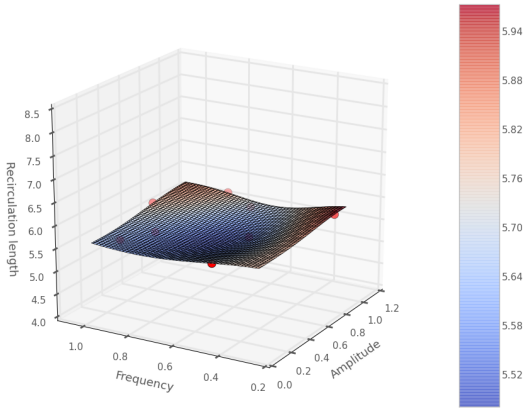


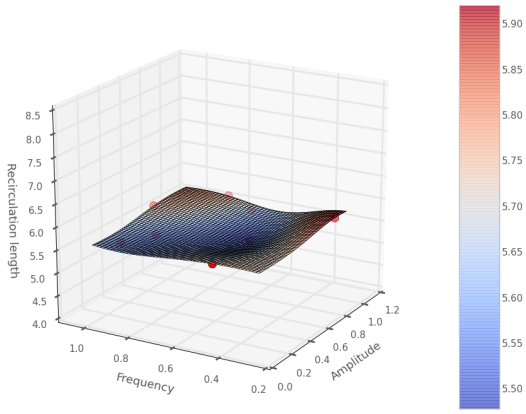
Figure 10.11: Grid convergence study for height different actuations (EASM closure). From top to bottom $(U_{jet}/U_{inf}, f_{jet}L_{ref}/U_{inf})$: (1.01, 0.087) (1.13, 0.23) (0.78, 0.29) (0.16, 0.14) (1.06, 0.20) (0.30, 0.27) (0.50, 0.25) (0.72, 0.16).



(a) Coarse grid.



(b) Medium grid.



(c) Fine grid.

Figure 10.12: Grid refinement effect on the recirculation length with kw-ne-easm turbulence closure and 256 time steps by periods.

To quantify the effects of the time step, three different discretization of 64, 128 and 256 time steps per period are selected, with the fine grid and a reduction of non linear residuals of 4 orders. Fig. 10.13 shows the evolution as the time step is reduced, for the finest grid and smallest dual time stepping order of convergence. The meta-models for

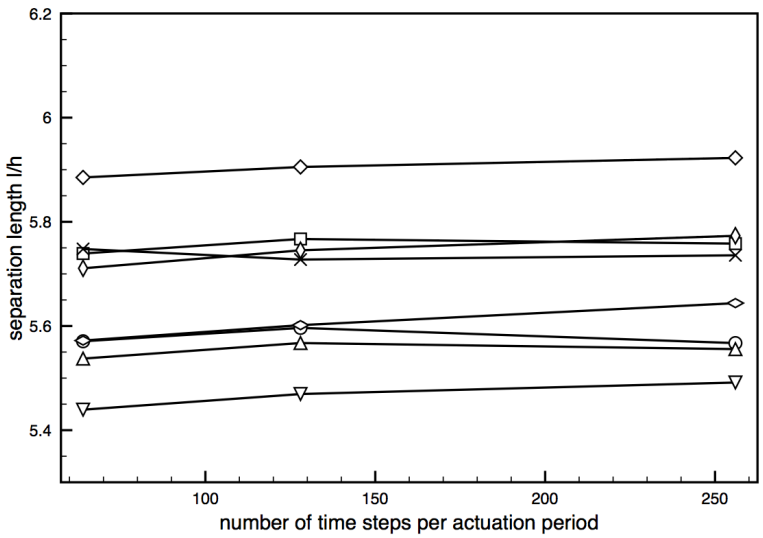
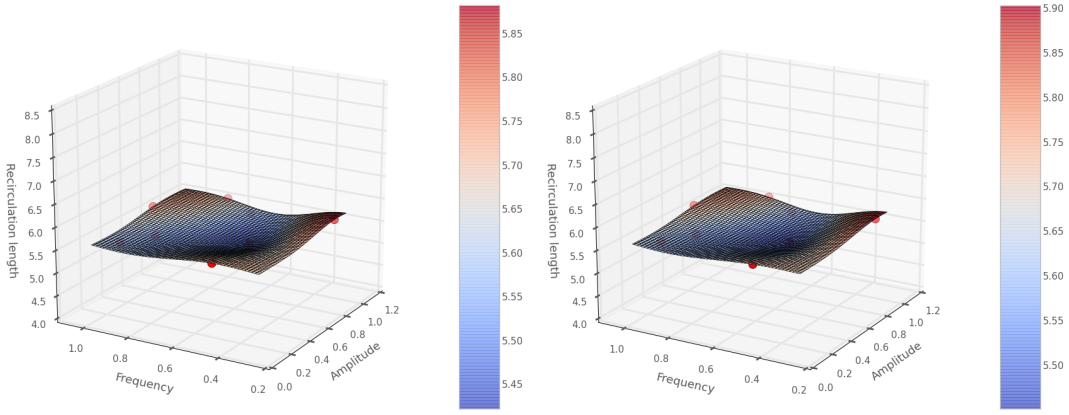


Figure 10.13: Time step convergence study for height different actuations (EASM closure). From top to bottom $(U_{jet}/U_{inf}, f_{jet}L_{ref}/U_{inf})$: (1.01, 0.087) (1.13, 0.23) (0.78, 0.29) (0.16, 0.14) (1.06, 0.20) (0.30, 0.27) (0.50, 0.25) (0.72, 0.16).

the three time steps are shown on Fig. 10.14.

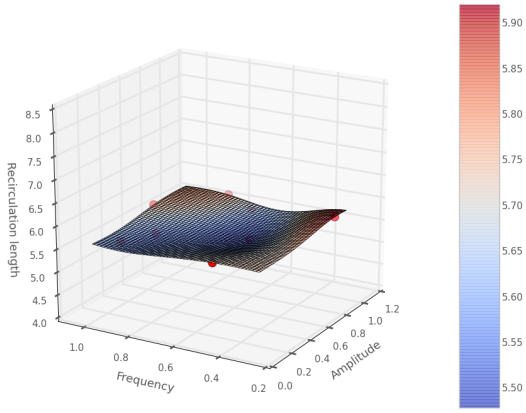
Dual time stepping method implies the convergence of the flow equations at each time step. The effect of the reduction of non linear residuals on the meta-model is quantified for 1, 2, 3 and 4 orders as stopping criterion. The medium grid, with 256 time steps per period is selected. The Fig. 10.15 shows the different meta-models obtained.

Height actuation configurations are selected randomly in the admissible domain. As can be seen, the medium grid provides satisfactory results: for all configurations except one, the separation length values evolve similarly, as the grid size is refined from the medium mesh to the fine mesh. Even if a full convergence of the value is not achieved, the ranking is correctly predicted. Regarding the time step selection, 64, 128 and 256 time steps per actuation period are tested. As observed, the recirculation length values also vary similarly as the time step is refined from 128 to 256 time steps per period, except for two configurations, generating a switch of two neighboring values. The overall meta-model shape is not really sensitive to grid and time step refinement as the effective control in medium frequencies and low frequencies is still located around the same location. The order of convergence has a biggest effect on the meta-model shape especially in high frequencies. With a low convergence of non linear residuals, the recirculation length is over-predicted.



(a) Large time step.

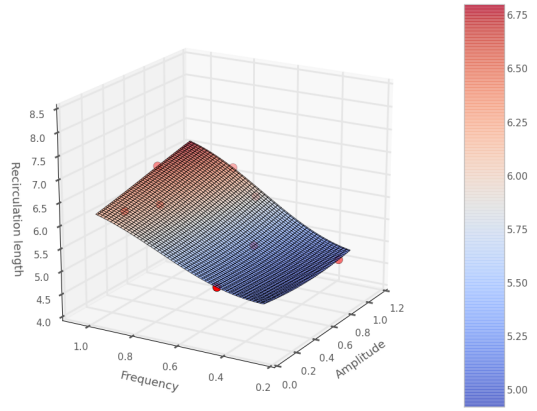
(b) Medium time step.



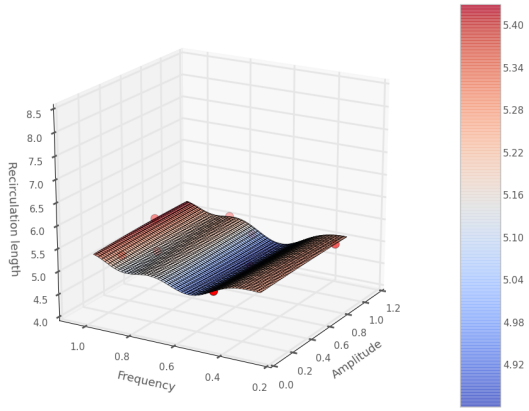
(c) Small time step.

Figure 10.14: Time step refinement effect on the recirculation length with k-ω-ne-easm turbulence closure and a fine grid and convergence of 4 orders of non linear residuals.

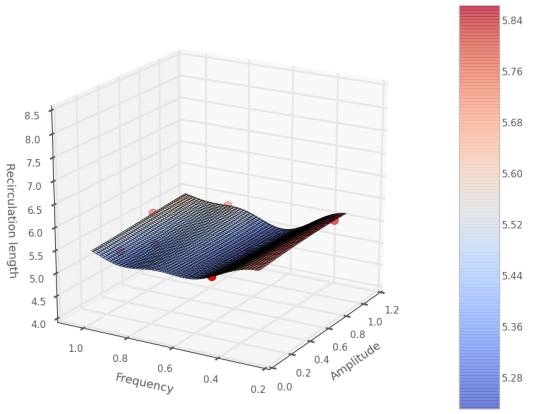
On this basis, the medium grid size, a 128 time steps per period and a reduction of non linear residuals of 3 orders are chosen for the optimization exercises. This choice is validated a posteriori: the improvement obtained during the optimization, as well as the discrepancy observed between the closures, are far larger than the discretization error reported here.



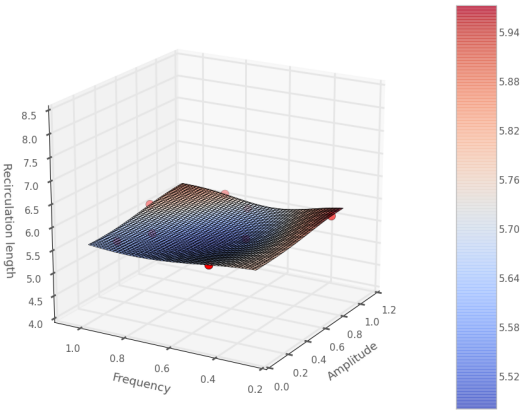
(a) Criterion reduction of 1 order.



(b) Criterion reduction of 2 orders.



(c) Criterion reduction of 3 orders.



(d) Criterion reduction of 4 orders.

Figure 10.15: Non linearities convergence criterion effect on the recirculation length with k-ω-ne-easm turbulence closure, a fine grid and a 256 time steps per period.

10.3 A naive optimization neglecting observation error

The objective of the optimization is the reduction of the separation length, by choosing efficient frequency and amplitude parameters. However, the estimation of this quantity cannot be more accurate than the local grid size. Therefore, a first optimization using the compressible solver neglecting observation error is first performed to understand the effect of neglecting geometrical considerations. To do so, the compressible solver is used on a mesh that contains 38235 nodes.Inlet boundary flow variables are determined by extracting the boundary layer from a flat plate computation at 105h from the leading edge. The inlet velocity profile is shown on Fig. 10.16 presents good agreements with the theory and the NASA CFL3D code. free-stream conditions based on Riemann Invariants are used, instead of no-slip conditions at the top boundary to reduce the mesh density.

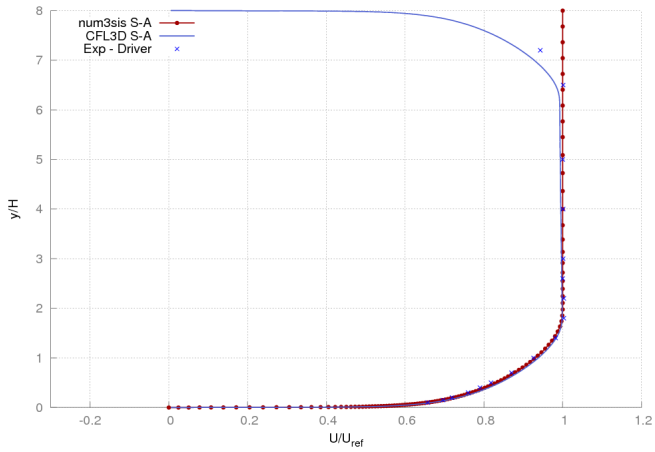
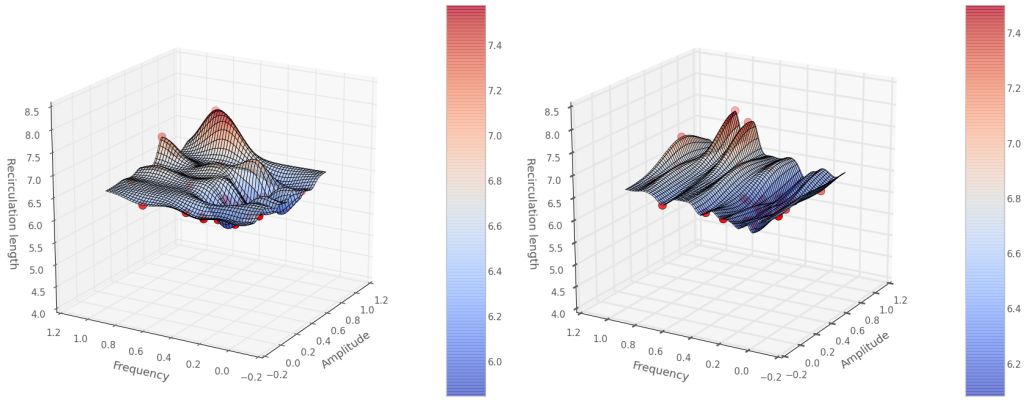


Figure 10.16: Inlet BFS velocity profile extracted at 105h from the leading edge of a flat plate.

The first 9 optimization steps are considered using the lower bound merit function, for the minimization of the BFS mean flow recirculation length with the compressible code. The turbulence closure is the Spalart-Allmaras model. The initial database contains 15 design points and, for each step, three simulations are run in parallel, so the final database contains 42 points in total. In this case, the recirculation length is computed using the closest grid point for which the friction coefficient vanishes. The mesh used is the coarsest one and the mesh size in the stream-wise direction in the recirculation area is $h_{coarse} = 1.10^{-3}m$, which gives a non dimensional size of $h^{adim}_{coarse} = 0.0787$. Fig. 10.17 shows the final meta-model and one iteration before the final meta-model obtained with the interpolating formulation. As seen the model oscillates with no real reason. Furthermore, a deeper look at the evaluation points shows that the optimization is not converging to a specific optimum indicating that the meta-model changes abruptly each time a new point is added to the database. This effect is seen where the last optimization iteration is removed from the final meta-model.



(a) Final interpolating meta-model.

(b) One optimization iteration before the final interpolating meta-model.

Figure 10.17: Left: final interpolating meta-model for the optimization of the recirculation length. Right: meta-model at one optimization iteration before the final interpolating meta-model for the optimization of the recirculation length. The solver is the compressible one with a Spalart-Allmaras closure. For the optimization, a lower bound merit function is used and the final database contains 42 points in total. The difference between the left and right meta-models enhances that the meta-model is not stable through optimization iterations.

This observation is similar to the one seen with the optimization validation over noisy analytical functions and with the NACA0015. It validates the necessity of considering the solver output as noisy and the optimization with noisy functions can be used here.

10.4 Impact of observation variance

The determination of the noise standard deviation still remains. The cost function is directly related to the mesh grid size. Indeed, the reattachment location is found by locating where the friction coefficient vanishes at the wall. The friction coefficient is interpolated between each nodes. The best linear interpolation error that can be done is in the order of h^{adim}_{coarse}.

The final database of the section 10.17 is reconsidered to construct non-interpolating meta-models with different level of error. Fig. 10.18 shows that the oscillating model becomes more reliable as the error standard deviation is increased. $3\tau_i \approx h^{adim}_{coarse} = 0.0787$ seems to be a threshold value, indicating that the error is effectively related to the cell size at reattachment point.

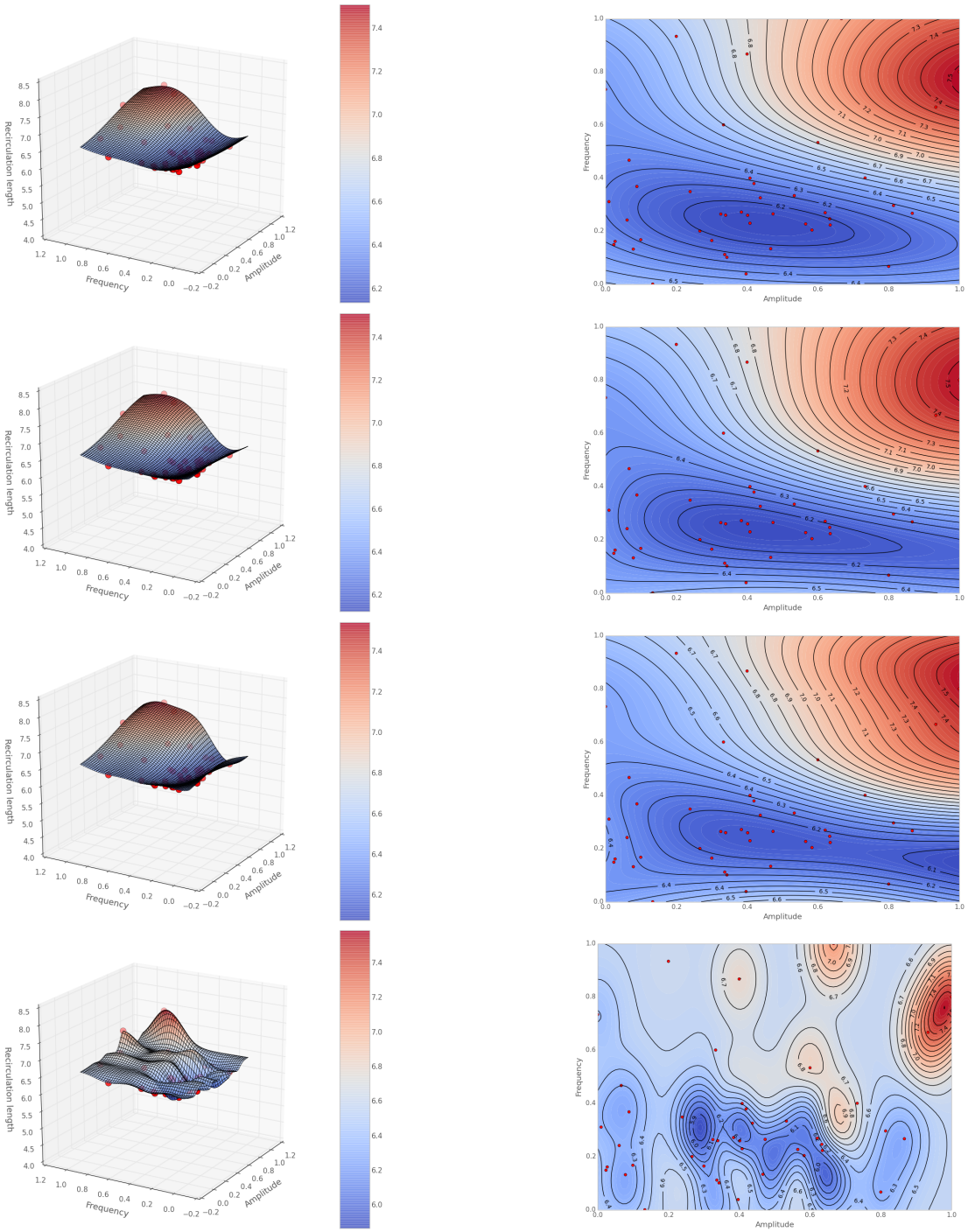


Figure 10.18: Comparison of meta-models using the same database coming from an optimization with different level of standard deviation to model the error. From the top to the bottom, $\tau_i = 0.1, \tau_i = 0.05, \tau_i = 0.02, \tau_i = 0.01$. As the standard deviation of the error decreases, the meta-model becomes unstable due to the presence of noise in the observations.

Point id	Amplitude (adim)	Frequency (adim)
0	1.0	0.4
1	0.4	0.8
2	0.0	0.6
3	0.2	0.2
4	0.6	0.0
5	0.8	1.0

Table 10.3: Initial database sampling for the optimization using the Backward Facing Step.

10.5 Impact of turbulence closures

It is well known that turbulence models have a huge impact on detached flow simulations. In [Garnier 2012], the drawn conclusion is that URANS turbulence models are not able to accurately reproduce the uncontrolled flow. Nevertheless, the capability to obtain similar trends for the optimum actuation is investigated. This study is carried out using the medium grid, 128 time steps per period and a reduction of non linear residuals of 3 orders, with the ISIS-CFD flow solver.

As described above, the evaluation of the separation length is achieved with great care. As the resulting noise in the functional evaluation may affect the Gaussian process model, the model construction includes a filtering process of amplitude $ls/2 = 0.01h$, where ls is the cell length at the re-attachment point. Note that this filtering process is far lower than the separation length value and, therefore, does not affect the results presented below.

The optimization of the synthetic jet parameters using four different turbulence models is performed. The chosen turbulence models are the Spalart-Allmaras (SA), k-ε Launder-Sharma (KE-LS), k-ω-SST Menter (KW-SST-MENTER) and quadratic explicit algebraic stress model based on k-ω (KW-NE-EASM).

The optimization procedure starts from a DOE phase, composed of only six points, that are selected according to a Latin Hypercube Sampling (LHS) [Sacks 1989]. This database is described on the Tab. 10.3 and represented on Fig. 10.19. Then, additional points are inserted in the Gaussian process model, according to the lower bound merit function. About ten enrichment steps are performed before convergence of the optimization procedure, corresponding to a total of about 40 simulations per optimization exercise.

10.5.1 Optimization results

The result of the previously described optimizations are found on the Fig. 10.20. It shows the Gaussian process models obtained using the different turbulence closures, as well as the configurations computed. As can be seen, the initial set of simulations covers

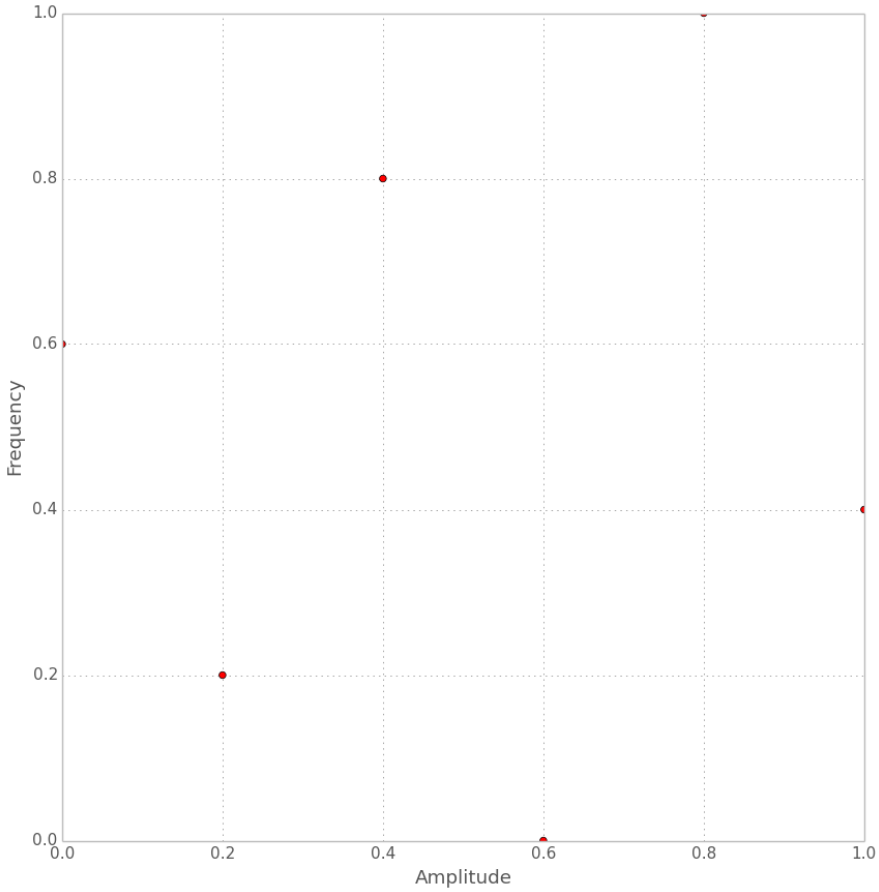


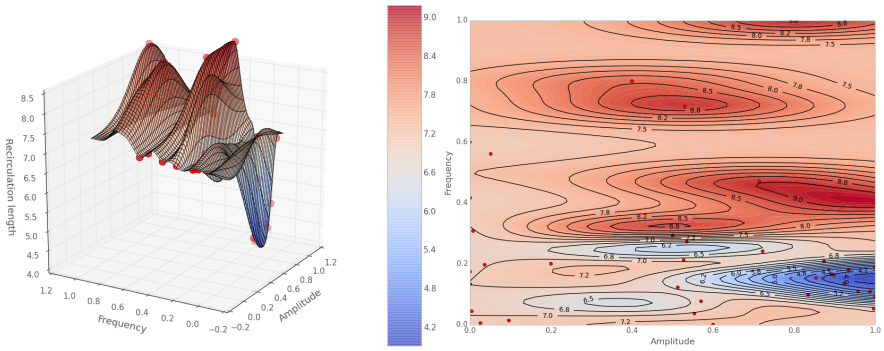
Figure 10.19: Initial database generated for the optimization of jet amplitude and frequency on the BFS.

the whole admissible domain, while the next evaluations converge towards the optimal parameters. At first, one can notice that the results based on the Spalart-Allmaras closure exhibit a behavior very different than other cases. In particular, the variations of the separation length with respect to the frequency are very irregular. The examination of the corresponding flows reveals that the Spalart-Allmaras closure predicts for some configurations a massive vortex shedding, which modifies abruptly the time-averaged separation length. The optimal configuration found is characterized by such a flow, as illustrated in Fig.10.21. The results obtained with the three other closures exhibit some similarities. In particular, only one minimum is identified by the Gaussian process models. For k-ε Launder-Sharma and EASM closures, it is located in the same region corresponding to medium actuation parameters. For the k-ω SST closure, the global shape of the response surface is similar, but the minimum corresponds to higher frequency and amplitude. One can underline that, for these three closures, the Gaussian process model exhibits a large flat region for which the actuation is efficient. Nevertheless, the precise values of the optimized control parameters are significantly different, as shown

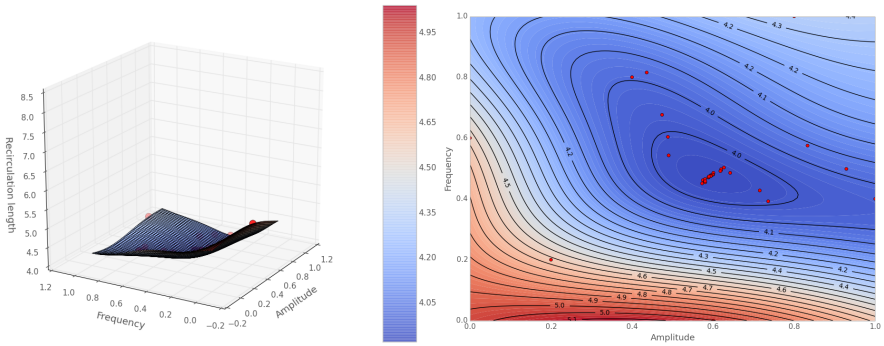
Turbulence closure	U_{jet}/U_{inf}	$f_{jet}L_{ref}/U_{inf}$	Length / Length ini
Spalart-Allmaras	1.13	0.059	0.69
k-ε	0.69	0.14	0.72
k-ω SST	0.40	0.24	0.84
EASM	0.56	0.19	0.70

Table 10.4: Optimum jet parameters found by the optimization over the Backward Facing Step after approximately 30 evaluations.

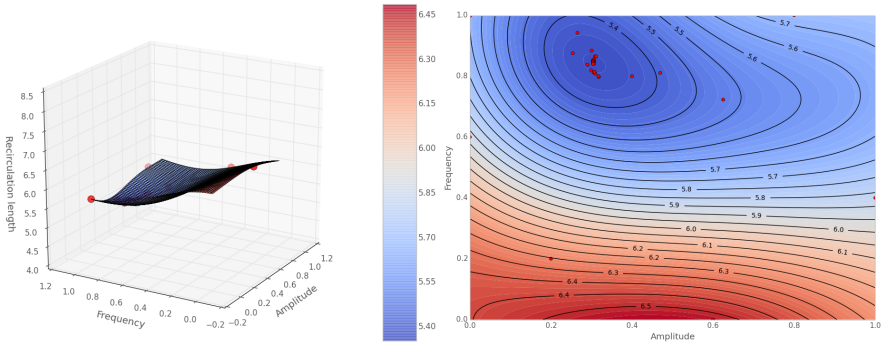
in Tab. 10.4.



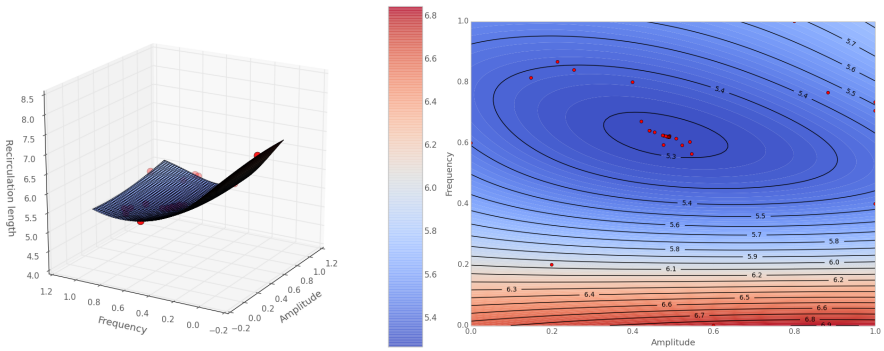
(a) Spalart-Allmaras closure.



(b) k-ε Launder-Sharma closure.



(c) k-ω-SST Menter closure.



(d) k-ω-ne-easm closure.

Figure 10.20: Gaussian process models of the separation length w.r.t. actuation frequency and amplitude, and simulated points.

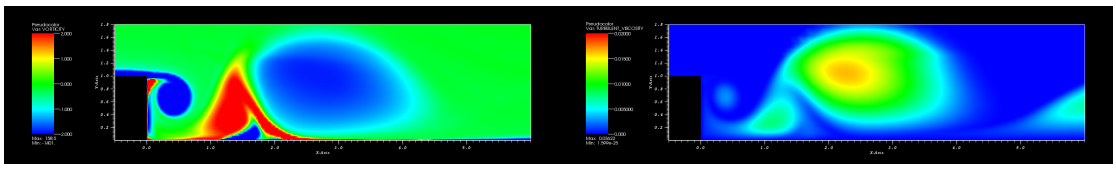
	Spalart-Allmaras	k-ε	k-ω SST	EASM
Optimized by Spal.-All.	**4.03**	4.24	5.98	5.75
Optimized by k-ε	8.79	**3.92**	5.73	5.33
Optimized by k-ω SST	8.32	4.03	**5.35**	5.34
Optimized by EASM	9.63	3.95	5.41	**5.27**

Table 10.5: Cross-comparison of recirculation lengths found for optimized configurations and evaluated using different closures.

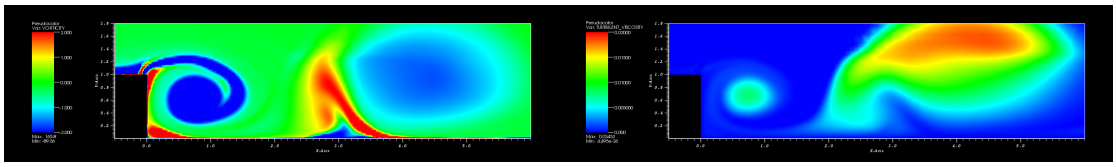
10.5.2 Cross-validation

A cross-validation exercise is then performed: the optimal configuration found by using a closure is simulated with other models. Results are shown in Tab. 10.5. One can note that the minimum value for each closure is obtained for the configuration optimized with the same closure. This is consistent, since the latter value is supposed to be the optimal one. The specific behavior of the Spalart-Allmaras closure can also be underlined: all the configurations found with other models exhibit a very bad performance value (far larger separation length) when evaluated using this closure. Beyond these observations regarding the best actuation found by the different closures, the following proposes to compare characteristics of the corresponding flows, to try to draw some conclusions. In this perspective, the instantaneous vorticity and turbulent viscosity fields are plotted for the best actuations found in Fig 10.21 to Fig.10.24. In each case, four snapshots are proposed, corresponding to the the zero blowing phase ($\phi = 0$), maximum blowing phase ($\phi = \pi/2$), zero suction phase ($\phi = \pi$) and maximum suction phase ($\phi = 3\pi/2$). The optimal configuration found by using the Spalart-Allmaras closure corresponds to a low frequency and large amplitude actuation, which generates a shedding of counter-rotative vortices. For the three other cases, the best actuation found is characterized by a higher frequency, yielding the birth of smaller vortices at the step corner. The three corresponding flows exhibit similar patterns, although the excitation frequency is different for each case. Especially, the most efficient flows predicted by the k-ω SST and the EASM closures are close to each other. Nevertheless, the non-linearities of the EASM closure generates a more vortical flow, with more intense variations of the turbulent viscosity. The best configuration found by using the k-ε Launder-Sharma closure has a lower actuation frequency. The corresponding flow exhibits vortices with larger span and a far higher turbulent viscosity level.

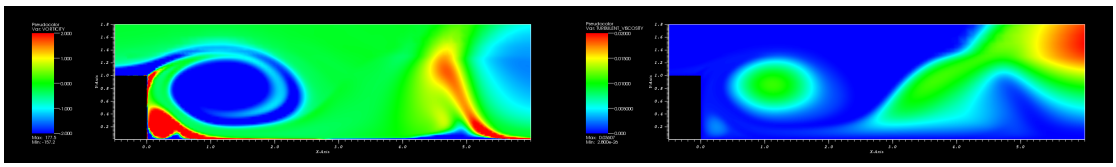
The proximity of the k-ω SST and the EASM closures is retrieved on the skin friction coefficient and velocity streamlines of the averaged flow figures shown on Fig. 10.25 and Fig. 10.26. The Spalart-Allmaras and k-ε Launder-Sharma closures exhibit a much different result with a sharper skin friction slope and a peak at $x/h \approx 5$. This exhibit the presence of a more intense recirculation bubble past the step.



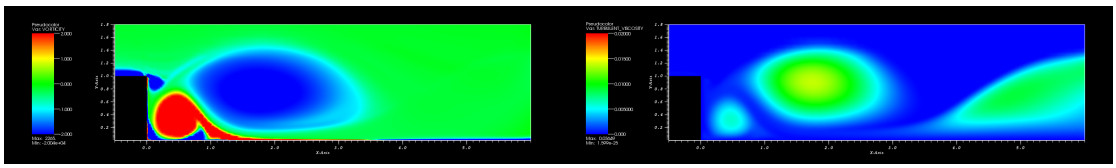
(a) $\Phi = 0$.



(b) $\Phi = \pi/2$.



(c) $\Phi = \pi$.



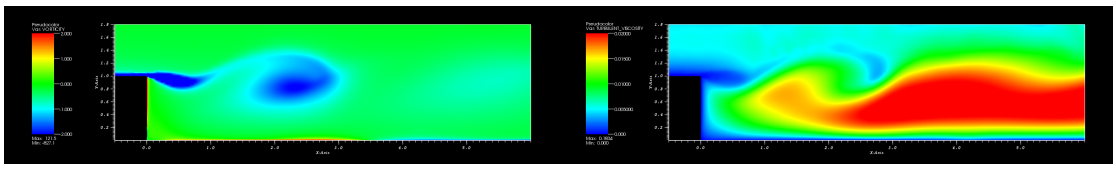
(d) $\Phi = 3\pi/2$.

Figure 10.21: Vorticity (left) and turbulent viscosity (right) fields for different actuation phases for control parameters optimized with Spalart-Allmaras closure (4.64, 0.014).

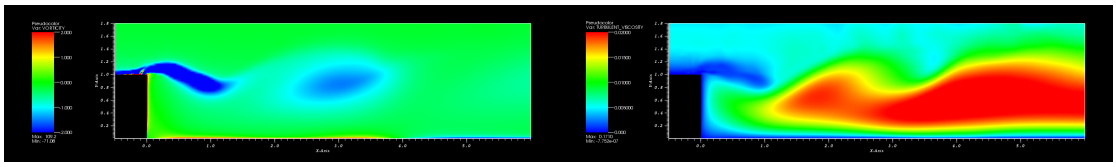
10.6 Discussion

For the test-case studied here, the characteristics of the baseline flow depend significantly on the turbulence closure. This is usual when separated flows are considered. The quantity of interest (separation length), which is also the optimization criterion for active control, exhibits variations of 25% when different closures are tested. This is obviously a difficult context for optimization. However, it is underlined that optimization could be achieved successfully, despite of these discrepancies, if the different closures provide similar flow changes, as the control parameters are modified. If the same error occurs during the whole optimization procedure, the different closures can yield the same optimum parameters, although the flow predictions differ. This has been already observed in shape optimization exercises related to pressure drop for instance: the optimum shape can be found even if the pressure field is not accurately predicted [Burman 2001].

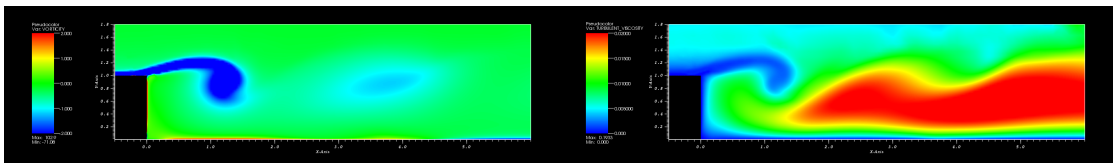
Unfortunately, the results presented above show that separated flows with actuation are not a so friendly context. The discrepancies observed for baseline flow yield different



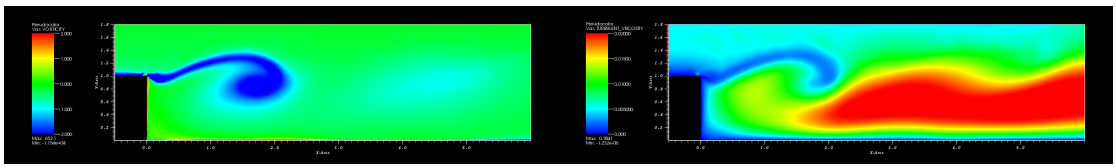
(a) $\Phi = 0$.



(b) $\Phi = \pi/2$.



(c) $\Phi = \pi$.

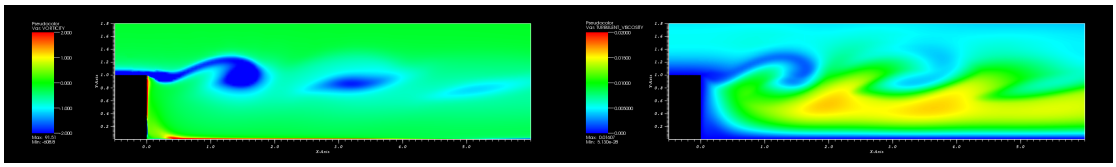


(d) $\Phi = 3\pi/2$.

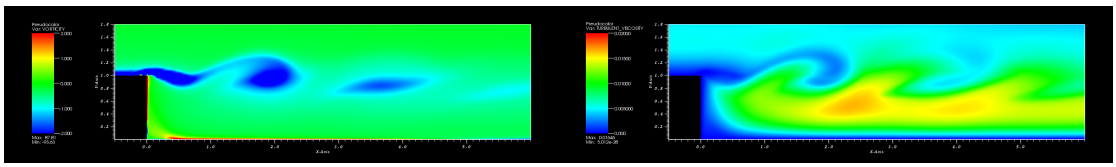
Figure 10.22: Vorticity (left) and turbulent viscosity (right) fields for different actuation phases for control parameters optimized with k-ε Launder-Sharma closure (11.13, 0.0089).

search directions and, finally, significantly different optimum parameters. One can conclude that RANS models cannot be employed blindly in a design optimization procedure to determine actuation parameters. Obviously, the "true" or physical optimal actuation parameters are unknown. For the baseline flow, the k-ω SST closure provides the best separation length prediction, according to the experiments. Nevertheless, it is even not clear if the best parameters predicted by this closure are close to the "true" optimum. The use of LES could provide some reference results, but optimization based on LES is still unaffordable, for computational time reasons.

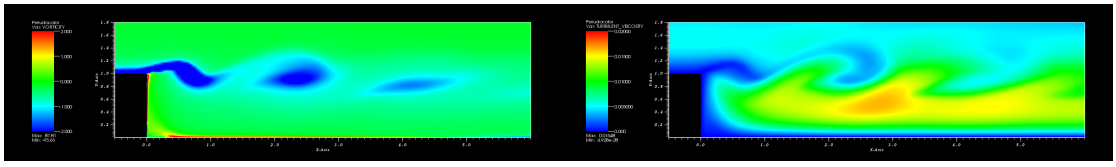
However, some similarities between the flows with optimal actuation have been reported, except for the Spalart-Allmaras closure. This seems to indicate that the different optimization exercises yield the same flow characteristics, whatever the closure. If this is confirmed, RANS models can still be used in a first design step, to determine the characteristics of efficient flow actuations.



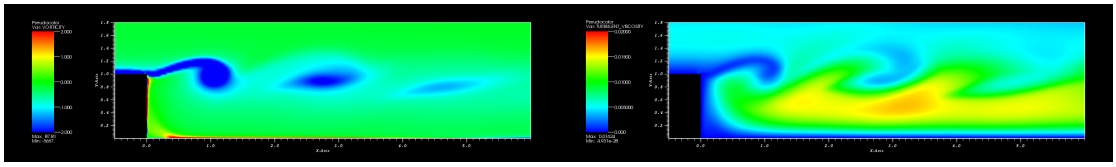
(a) $\Phi = 0$.



(b) $\Phi = \pi/2$.

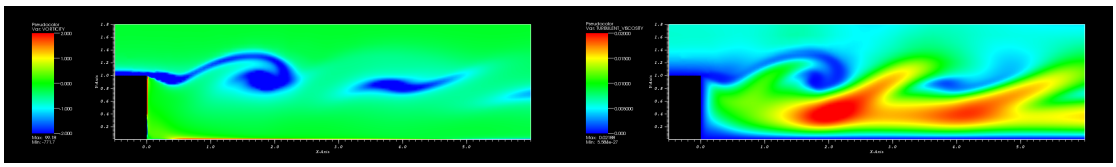


(c) $\Phi = \pi$.

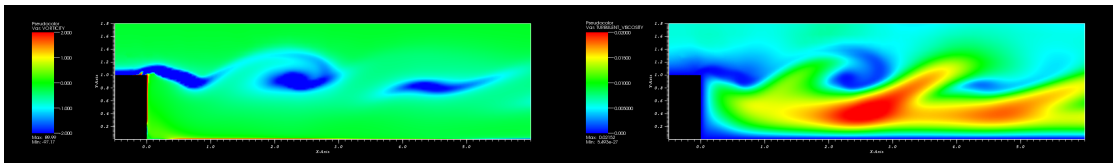


(d) $\Phi = 3\pi/2$.

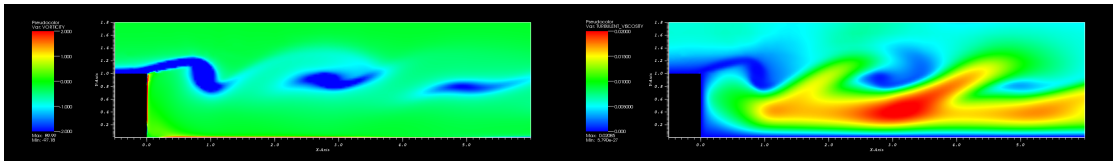
Figure 10.23: Vorticity (left) and turbulent viscosity (right) fields for different actuation phases for control parameters optimized with k-ω SST Menter closure (18.73, 0.0052).



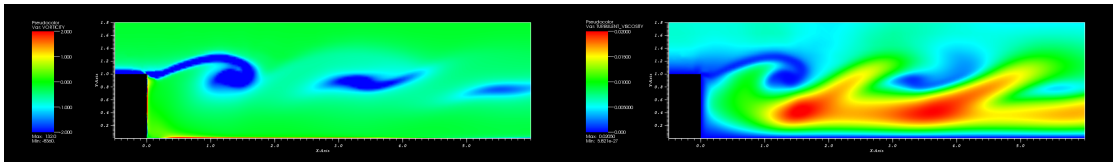
(a) $\Phi = 0$.



(b) $\Phi = \pi/2$.



(c) $\Phi = \pi$.



(d) $\Phi = 3\pi/2$.

Figure 10.24: Vorticity (left) and turbulent viscosity (right) fields for different actuation phases for control parameters optimized with EASM closure (14.77, 0.0072).

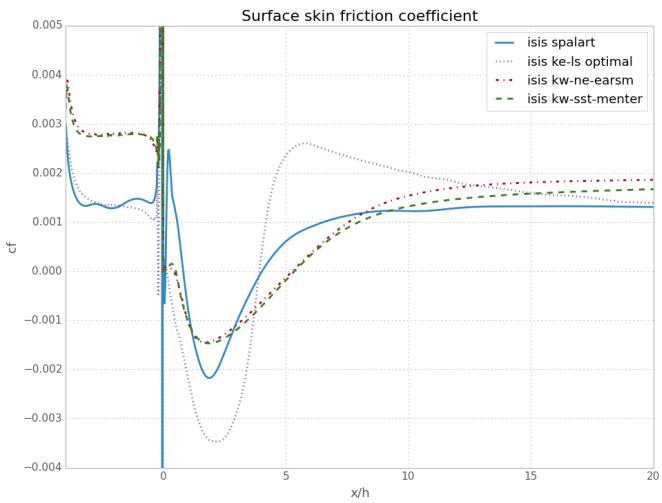
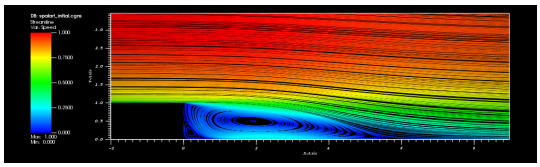
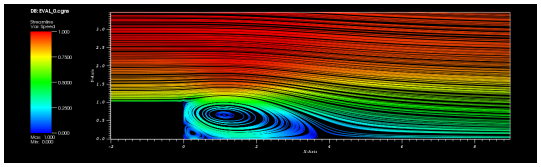


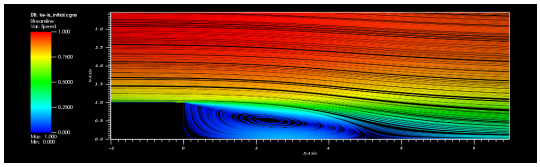
Figure 10.25: Skin friction coefficients of the time-averaged flow for the best actuation parameters with each turbulence closures.



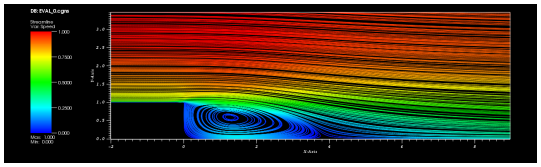
(a) Spalart-Allmaras closure.



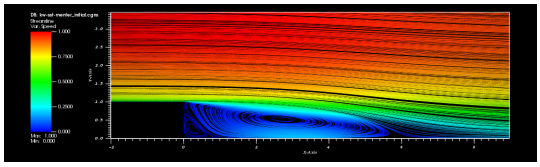
(b) Spalart-Allmaras closure - controlled.



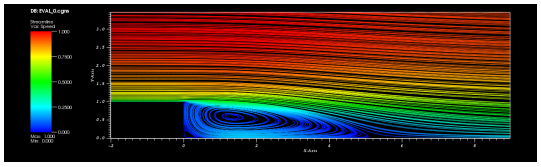
(c) k-ε Launder-Sharma closure.



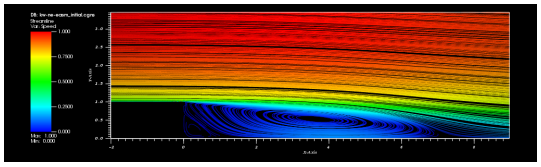
(d) k-ε Launder-Sharma closure - controlled.



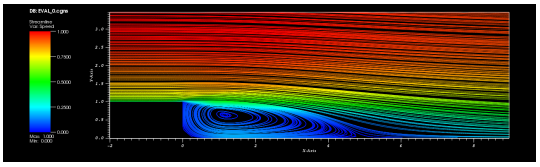
(e) k-ω-SST Menter closure.



(f) k-ω-SST Menter closure - controlled.



(g) k-ω-ne-easm closure.



(h) k-ω-ne-easm closure - controlled.

Figure 10.26: Comparison of velocity streamlines for different turbulence closures for the time-averaged flow over the backward facing step without control (left) and with optimized flow control (right).

Part V

Conclusion and prospects

In this thesis, a combination of numerical methods has been presented to tackle the optimization of active control devices for separated turbulent flows. This problem is hard in essence as it copes with various disciplines, such as numerical simulation, turbulence modeling, high performance computing, optimization, etc. Limitations and difficulties to apply some classical optimization methods, as gradient based or heuristic methods have been reported. They are mainly due to multi-modality, unsteadiness leading to cumbersome cost function gradient evaluation using adjoint and CPU time restrictions.

Coupling optimization with unsteady turbulent flow simulations is a big challenge as it introduces stiff requirements and constraints in terms of optimization techniques and physical modeling. The choice has been focused on RANS models, due to CPU cost, and on meta-model based optimization, for its efficiency and global search capability.

In this thesis, computational methods for the simulation of unsteady compressible turbulent flows followed by optimization methods based on meta-models have been first described. Observation variance was introduced in the meta-model generation to overcome cost function observation uncertainties. All the implemented methods have been carefully tested on a certain number of test cases relevant to the considered problems, before considering two optimization test cases: the NACA0015 and the Backward Facing Step.

The Efficient Global Optimization strategy applied to the maximization NACA0015 airfoil lift at 18 degrees and to the reduction of the recirculation length past a Backward Facing Step was really effective to find synthetic jet actuator parameters optima. In both cases, the optimization exercises were carried out using different RANS turbulence closures. For the NACA0015, an increase in lift of approximately 20% could be achieved for both the Spalart-Allmaras and k-ω SST turbulence closures after approximately 15 solver evaluations. The reduction of the recirculation length, past the Backward Facing Step, achieved by the optimizer was on average 73% of the initial recirculation length after 30 optimization evaluations using the Spalart-Allmaras, k-ε Launder-Sharma, k-ω SST Menter and k-ω-ne-easm turbulence closure.

These optimization exercises showed the necessity to account for numerical errors during the meta-model based optimization. Indeed, the use of interpolating models yields oscillating and unrealistic representations, leading the optimization process to failure. The introduction of the noisy observation concept allowed to overcome this difficulty. The numerical error was identified to come from various stages of the simulation process, such as the grid, the reduction order of the residuals, etc.

Apart from the uncertainty introduced by numerical errors, it has been found that the turbulence modeling has the greatest impact, yielding to significantly different optimum parameters. Furthermore, the quantities of interest could exhibit variations of 25% when different closures were tested. Despite of these discrepancies, some similarities between the flows with optimum actuation have been reported, for both the control over the NACA0015 and the Backward Facing Step, indicating that RANS models could still be employed to determine ranges of effective flow parameters.

There are still a lot of open questions for optimization based on unsteady turbulent flows to be extended to more complex industrial cases.

The first one concerns the optimization point of view and the error management. Indeed, it has been shown that estimating the error was necessary to conduct a successful optimization in this context. Estimating the error was done approximately by increasing the level of variance until a satisfactory meta-model was obtained. This aspect is certainly the most complex problem as one has to derive a specific approach to properly estimate the error. To handle more than two parameters, having an automatic way to tune the observation variance, such as in [J. Forrester 2006], is even more necessary as there is no possibility to see the meta-model.

Another perspective would concern the account of physical uncertainties during the optimization. Indeed, the practical use of actuators will be subject to several uncertainties, arising from flow conditions (Reynolds number, Mach number, incidence, etc.). In this context, a robust optimization of actuator parameters should be achieved, which consist in minimizing the expectation of the cost function. For the EGO method, a promising approach has been proposed in [Janusevskis 2013].

The second open question is to overcome limitations introduced by RANS modeling. Extending the method to hybrid RANS/LES models is straightforward from computational point of view, but yield an explosion of CPU time. A good candidate in the perspective of solving this ambitious problem, is to use partially converged solutions [Picheny 2012] to loosen a bit the requirements. Of course, this can only be done if the first point (variance management) is solved. A final perspective would be to have an adaptive algorithm, that automatically sets the required simulation length, and mixes short simulations with high variance and long simulations with small variance [Picheny 2013].

The work performed during this thesis has set up the foundations to consider a large panel of new applications. In a flow simulation point of view, the extension to 3D cases optimization is of course one of the first exercises that would really make sense for the industry. Then, the extension of optimization with large number of parameters to be optimized might be a step forward to solve realistic problems. That could include the position of the actuator, its orientation, etc. Finally, other actuators than synthetic jets could be considered. Indeed, a large number of experimental and numerical reports are available using alternative methods, such as DBD plasma actuators, or pulsated jets.

Appendix A

Fluxes Jacobians

A.1 Inviscid Jacobians

A.1.1 General problem of finding the Jacobian for a numerical inviscid flux

A numerical inviscid flux can be expressed as:

$$\boldsymbol{\Phi_{ij}} = \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}) \tag{A.1}$$

If $\boldsymbol{W_i^c}$, $\boldsymbol{W_i^c}$ and $\boldsymbol{\eta_{ij}}$ are independent, the Jacobian of a numerical inviscid flux with respect to the conservative state $\boldsymbol{W_i^c}$ can be expressed as:

$$\frac{\partial \boldsymbol{\Phi_{ij}}}{\partial \boldsymbol{W_i^c}} = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_i^c}} \tag{A.2}$$

The same equation holds for the Jacobian with respect to the conservative state $\boldsymbol{W_j^c}$.

In the case the conservative state $\boldsymbol{W_j^c}$ is expressed as $\boldsymbol{W_j^c}(\boldsymbol{W_i^c})$, then the numerical inviscid flux can be expressed as $\boldsymbol{\Phi_{ij}} = \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}})$ and its Jacobian slightly differs:

$$\frac{\partial \boldsymbol{\Phi_{ij}}}{\partial \boldsymbol{W_i^c}} = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_i^c}} + \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_j^c}} \frac{\partial \boldsymbol{W_j^c}}{\partial \boldsymbol{W_i^c}} \tag{A.3}$$

The same holds for the Jacobian with respect to the conservative state $\boldsymbol{W_j^c}(\boldsymbol{W_i^c})$.

A.1.1.1 Application to the Rusanov flux

The spectral radius of a square matrix A is denoted by $\rho\,(A)$ and defined by:

$$\rho\,(A) \stackrel{\text{def}}{=} \max_i(|\lambda_i|)$$

with $\lambda_1, ..., \lambda_i$ the eigenvalues of A.

The Rusanov flux is defined by:

$$\boldsymbol{\Phi_{ij}}^{RUS} = \frac{1}{2}(\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}))\boldsymbol{\eta_{ij}} - \frac{1}{2}d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}) \tag{A.4}$$

with the Rusanov numerical dissipation $d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}) = \max_{k\in\{i,j\}} \rho\left(\mathcal{A}(\boldsymbol{W_k^c}, \boldsymbol{\eta_{ij}})\right)(\boldsymbol{W_j^c} - \boldsymbol{W_i^c})$ and with $\mathcal{A}(\boldsymbol{W_k^c}, \boldsymbol{\eta_{ij}}) = \frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_k^c})}{\partial \boldsymbol{W_k^c}}$

The Rusanov flux Jacobian calculated here is done by doing the assumption of a frozen flux Jacobian matrix $\widetilde{\mathcal{A}}$ with respect to $\boldsymbol{W_i^c}$ and $\boldsymbol{W_j^c}$. This assumption have demonstrated to give good results in [Batten 1997b].

Furthermore, it is used here only to demonstrate the comparative results between the Jacobian for any numerical inviscid flux and the Jacobian of the Rusanov numerical flux itself with no loss of generality.

Jacobian with assumption of $\boldsymbol{W_i^c}$ and $\boldsymbol{W_j^c}$ to be independant

$$\begin{aligned}
\frac{\partial \boldsymbol{\Phi_{ij}}^{RUS}}{\partial \boldsymbol{W_i^c}} &= \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_i^c}} \\
&= \frac{\partial(\frac{1}{2}(\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}))\boldsymbol{\eta_{ij}} - \frac{1}{2}d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}))}{\partial \boldsymbol{W_i^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_i^c}} - \frac{1}{2}\frac{\partial d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_i^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_i^c}} + \frac{1}{2}\boldsymbol{I_n} \max_{k\in\{i,j\}} \rho\left(\widetilde{\mathcal{A}_k}\right)
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
\frac{\partial \boldsymbol{\Phi_{ij}}^{RUS}}{\partial \boldsymbol{W_j^c}} &= \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_j^c}} \\
&= \frac{\partial(\frac{1}{2}(\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}))\boldsymbol{\eta_{ij}} - \frac{1}{2}d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}))}{\partial \boldsymbol{W_j^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_j^c}} - \frac{1}{2}\frac{\partial d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_j^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_j^c}} - \frac{1}{2}\boldsymbol{I_n} \max_{k\in\{i,j\}} \rho\left(\widetilde{\mathcal{A}_k}\right)
\end{aligned} \tag{A.6}$$

Jacobian with assumption of $\boldsymbol{W_j^c}(\boldsymbol{W_i^c})$

$$
\begin{aligned}
\frac{\partial \boldsymbol{\Phi_{ij}}^{RUS}}{\partial \boldsymbol{W_i^c}} &= \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_i^c}} \\
&= \frac{\partial(\frac{1}{2}(\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}(\boldsymbol{W_i^c})))\boldsymbol{\eta_{ij}} - \frac{1}{2}d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}}))}{\partial \boldsymbol{W_i^c}} \\
&= \frac{1}{2}\left(\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_i^c}} + \frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}(\boldsymbol{W_i^c}))\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_j^c}}\frac{\partial \boldsymbol{W_j^c}}{\partial \boldsymbol{W_i^c}}\right) - \frac{1}{2}\frac{\partial d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}(\boldsymbol{W_i^c}), \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_i^c}} \\
&= \frac{1}{2}\left(\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_i^c}} + \boldsymbol{I_n}\max_{k\in\{i,j\}}\rho\left(\widetilde{\mathcal{A}_k}\right)\right) + \frac{1}{2}\left(\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}(\boldsymbol{W_i^c}))\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_j^c}} - \boldsymbol{I_n}\max_{k\in\{i,j\}}\rho\left(\widetilde{\mathcal{A}_k}\right)\right)\frac{\partial \boldsymbol{W_j^c}}{\partial \boldsymbol{W_i^c}} \\
&= \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_i^c}} + \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_j^c}}\frac{\partial \boldsymbol{W_j^c}}{\partial \boldsymbol{W_i^c}}
\end{aligned}
\tag{A.7}
$$

$$
\begin{aligned}
\frac{\partial \boldsymbol{\Phi_{ij}}^{RUS}}{\partial \boldsymbol{W_j^c}} &= \frac{\partial(\frac{1}{2}(\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c}) + \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c}))\boldsymbol{\eta_{ij}} - \frac{1}{2}d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}}))}{\partial \boldsymbol{W_j^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c})}{\partial \boldsymbol{W_j^c}}\boldsymbol{\eta_{ij}} - \frac{1}{2}\frac{\partial d_{RUS}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})}{\partial \boldsymbol{W_j^c}} \\
&= \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_j^c})}{\partial \boldsymbol{W_j^c}}\boldsymbol{\eta_{ij}} - \frac{1}{2}\boldsymbol{I_n}\max_{k\in\{i,j\}}\rho\left(\widetilde{\mathcal{A}_k}\right) \\
&= \frac{\partial \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})^{RUS}}{\partial \boldsymbol{W_j^c}}
\end{aligned}
\tag{A.8}
$$

The same approach holds for the Jacobian with assumption of $\boldsymbol{W_i^c}(\boldsymbol{W_j^c})$.

A.1.2 Inviscid flux Jacobian

The inviscid flux on a facet is defined by:

$$
\boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}} = \boldsymbol{F}(\boldsymbol{W_i^c})\eta_{ij_x} + \boldsymbol{G}(\boldsymbol{W_i^c})\eta_{ij_y} + \boldsymbol{H}(\boldsymbol{W_i^c})\eta_{ij_z} = \begin{pmatrix} \rho(\boldsymbol{V}\cdot\boldsymbol{\eta_{ij}}) \\ \rho u(\boldsymbol{V}\cdot\boldsymbol{\eta_{ij}}) + p\eta_{ij_x} \\ \rho v(\boldsymbol{V}\cdot\boldsymbol{\eta_{ij}}) + p\eta_{ij_y} \\ \rho w(\boldsymbol{V}\cdot\boldsymbol{\eta_{ij}}) + p\eta_{ij_z} \\ (\rho E + p)\boldsymbol{V}\cdot\boldsymbol{\eta_{ij}} \end{pmatrix}
\tag{A.9}
$$

The inviscid flux Jacobian on a facet is defined by:

$$\frac{\partial \boldsymbol{\mathcal{F}}(\boldsymbol{W_i^c})\boldsymbol{\eta_{ij}}}{\partial \boldsymbol{W_j^c}} = \begin{pmatrix} 0 & \eta_{ij_x} & \eta_{ij_y} & \eta_{ij_z} & 0 \\ \eta_{ij_x}a_0 - uV_{\boldsymbol{\eta_{ij}}} & V_{\boldsymbol{\eta_{ij}}} - a_3\eta_{ij_x}u & \eta_{ij_y}u - a_2\eta_{ij_x}v & \eta_{ij_z}u - a_2\eta_{ij_x}w & a_2\eta_{ij_x} \\ \eta_{ij_y}a_0 - vV_{\boldsymbol{\eta_{ij}}} & \eta_{ij_x}v - a_2\eta_{ij_y}u & V_{\boldsymbol{\eta_{ij}}} - a_3\eta_{ij_y}v & \eta_{ij_z}v - a_2\eta_{ij_y}w & a_2\eta_{ij_y} \\ \eta_{ij_z}a_0 - wV_{\boldsymbol{\eta_{ij}}} & \eta_{ij_x}w - a_2\eta_{ij_z}u & \eta_{ij_y}w - a_2\eta_{ij_z}v & V_{\boldsymbol{\eta_{ij}}} - a_3\eta_{ij_z}w & a_2\eta_{ij_z} \\ V_{\boldsymbol{\eta_{ij}}}(a_0 - a_1) & \eta_{ij_x}a_1 - a_2uV_{\boldsymbol{\eta_{ij}}} & \eta_{ij_y}a_1 - a_2vV_{\boldsymbol{\eta_{ij}}} & \eta_{ij_z}a_1 - a_2wV_{\boldsymbol{\eta_{ij}}} & \gamma V_{\boldsymbol{\eta_{ij}}} \end{pmatrix} \tag{A.10}$$

with:

$$a_0 = \frac{1}{2}(\gamma - 1)\boldsymbol{V}\cdot\boldsymbol{V}, \quad a_1 = \gamma E - a_0, \quad a_2 = \gamma - 1$$

$$a_3 = \gamma - 2, \quad V_{\boldsymbol{\eta_{ij}}} = \boldsymbol{V}\cdot\boldsymbol{\eta_{ij}}$$

A.2 Boundary Jacobians

The purpose of this section is to describe the calculation of the Jacobian $\frac{\partial \boldsymbol{W_b^c}}{\partial \boldsymbol{W_i^c}}$. Here, the boundary state is denoted by b and the interior state by i.

A.2.1 Decomposition of the Jacobian

In our framework, the variable used are mainly the primitive variable $\boldsymbol{W^p}$ even if the final update is computed in a conservative form. The main goal is to have the different Jacobians expressed in conservative variables. The Jacobian $\frac{\partial \boldsymbol{W_b^c}}{\partial \boldsymbol{W_i^c}}$ can be decomposed with:

$$\frac{\partial \boldsymbol{W_b^c}}{\partial \boldsymbol{W_i^c}} = \frac{\partial \boldsymbol{W_b^c}}{\partial \boldsymbol{W_b^p}}\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}\frac{\partial \boldsymbol{W_i^p}}{\partial \boldsymbol{W_i^c}} \tag{A.11}$$

With this decomposition, the only part that depends on the boundary condition type is $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$. The other two transformation Jacobians are detailed below.

A.2.1.1 Computation of conservative over primitive variables Jacobian

The transformation from primitive to conservative variables can be expressed as:

$$\boldsymbol{W^c} = \begin{pmatrix} \boldsymbol{W^p}[0] \\ \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[1] \\ \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[2] \\ \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[3] \\ \frac{\boldsymbol{W^p}[4]}{\gamma-1} + \frac{1}{2}\frac{(\boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[1])^2+(\boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[2])^2+(\boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[3])^2}{\boldsymbol{W^p}[0]} \end{pmatrix} \tag{A.12}$$

Thus, the Jacobian $\frac{\partial \boldsymbol{W^c}}{\partial \boldsymbol{W^p}}$ reads:

$$\frac{\partial \boldsymbol{W^c}}{\partial \boldsymbol{W^p}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \boldsymbol{W^p}[1] & \boldsymbol{W^p}[0] & 0 & 0 & 0 \\ \boldsymbol{W^p}[2] & 0 & \boldsymbol{W^p}[0] & 0 & 0 \\ \boldsymbol{W^p}[3] & 0 & 0 & \boldsymbol{W^p}[0] & 0 \\ \frac{\boldsymbol{W^p}[1]^2+\boldsymbol{W^p}[2]^2+\boldsymbol{W^p}[3]^2}{2} & \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[1] & \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[2] & \boldsymbol{W^p}[0]\cdot\boldsymbol{W^p}[3] & \frac{1}{\gamma-1} \end{pmatrix} \tag{A.13}$$

A.2.1.2 Computation of primitive over conservative variables Jacobian

The transformation from conservative to primitive variables can be expressed as:

$$\boldsymbol{W^p} = \begin{pmatrix} \boldsymbol{W^c}[0] \\ \frac{\boldsymbol{W^c}[1]}{\boldsymbol{W^c}[0]} \\ \frac{\boldsymbol{W^c}[2]}{\boldsymbol{W^c}[0]} \\ \frac{\boldsymbol{W^c}[3]}{\boldsymbol{W^c}[0]} \\ (\gamma-1)\cdot(\boldsymbol{W^c}[4]-\frac{1}{2}\cdot\boldsymbol{W^c}[0]\cdot((\frac{\boldsymbol{W^c}[1]}{\boldsymbol{W^c}[0]})^2+(\frac{\boldsymbol{W^c}[2]}{\boldsymbol{W^c}[0]})^2+(\frac{\boldsymbol{W^c}[3]}{\boldsymbol{W^c}[0]})^2)) \end{pmatrix} \tag{A.14}$$

Thus, the Jacobian $\frac{\partial \boldsymbol{W^p}}{\partial \boldsymbol{W^c}}$ reads:

$$\frac{\partial \boldsymbol{W^p}}{\partial \boldsymbol{W^c}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{\boldsymbol{W^p}[1]}{\boldsymbol{W^p}[0]} & \frac{1}{\boldsymbol{W^p}[0]} & 0 & 0 & 0 \\ -\frac{\boldsymbol{W^p}[2]}{\boldsymbol{W^p}[0]} & 0 & \frac{1}{\boldsymbol{W^p}[0]} & 0 & 0 \\ -\frac{\boldsymbol{W^p}[3]}{\boldsymbol{W^p}[0]} & 0 & 0 & \frac{1}{\boldsymbol{W^p}[0]} & 0 \\ \frac{1}{2}\cdot\gamma_1\cdot(\boldsymbol{W^p}[1]^2+\boldsymbol{W^p}[2]^2+\boldsymbol{W^p}[3]^2) & -\gamma_1\cdot\boldsymbol{W^p}[1] & -\gamma_1\cdot\boldsymbol{W^p}[2] & -\gamma_1\cdot\boldsymbol{W^p}[3] & \gamma_1 \end{pmatrix} \tag{A.15}$$

with $\gamma_1 = \gamma - 1$.

A.2.2 Symmetry condition

In the case of a symmetry condition, the boundary state is computed using the interior state only. For our computation, it is computed in a weak form by setting an opposite tangential velocity at the boundary state.

Let us define the boundary state vector by $\boldsymbol{W_b^p}$ a function of the internal state $\boldsymbol{W_b^p}(\boldsymbol{W_i^p})$.

The vector $\boldsymbol{W_b^p}$ is defined by:

$$\boldsymbol{W_b^p} = \begin{pmatrix} \boldsymbol{W_i^p}[0] \\ \boldsymbol{V}_i - 2(\boldsymbol{V}_i \cdot \widehat{\boldsymbol{n}})\widehat{\boldsymbol{n}} \\ \boldsymbol{W_i^p}[4] \end{pmatrix} \tag{A.16}$$

with $\widehat{\boldsymbol{n}}$ a unit vector normal to the boundary pointing outwards the domain and $\boldsymbol{V}_i$ the internal velocity vector of the flow.

The Jacobian matrix $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$ reads:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1-\widehat{n}_x^2 & \widehat{n}_y\widehat{n}_z & \widehat{n}_z\widehat{n}_x & 0 \\ 0 & \widehat{n}_x\widehat{n}_y & 1-\widehat{n}_y^2 & \widehat{n}_z\widehat{n}_y & 0 \\ 0 & \widehat{n}_z\widehat{n}_x & \widehat{n}_z\widehat{n}_y & 1-\widehat{n}_z^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.17}$$

A.2.3 Supersonic inlet

In the case of a supersonic inlet, $\boldsymbol{W_b^p}(\boldsymbol{W_\infty^p}) = \boldsymbol{W_\infty^p}$ with the freestream state denoted by $\boldsymbol{W_\infty^p}$.

Thus, the Jacobian $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$ reads:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}} = \boldsymbol{0} \tag{A.18}$$

A.2.4 Supersonic outlet

In the cas of a supersonic outlet, $\boldsymbol{W_b^p}(\boldsymbol{W_i^p}) = \boldsymbol{W_i^p}$.

Thus, the Jacobian $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$ reads:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}} = \boldsymbol{I_n} \tag{A.19}$$

A.2.5 Riemann Invariant subsonic inlet

Let us define the Riemann Invariant vector by $\boldsymbol{W^R}$. The boundary state $\boldsymbol{W_b^p}$ is a function of the Riemann Invariant at the boundary $\boldsymbol{W_b^R}(\boldsymbol{W_i^p}, \boldsymbol{W_\infty^p})$ that depends on the interior state $\boldsymbol{W_i^p}$ and the freestream state $\boldsymbol{W_\infty^p}$. In other words:

$$\boldsymbol{W_b^p} = \boldsymbol{W_b^p}(\boldsymbol{W_b^R}(\boldsymbol{W_i^p}, \boldsymbol{W_\infty^p})) \tag{A.20}$$

The Jacobian $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$ is:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}} = \frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}} \frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}} \tag{A.21}$$

The Riemann Invariant $\boldsymbol{W^R}$ contains the following variables:

$$\boldsymbol{W^R} = \begin{pmatrix} \mathcal{R}^- \\ S \\ \boldsymbol{V_{\widehat{\tau}}} \\ \mathcal{R}^+ \end{pmatrix} = \begin{pmatrix} \boldsymbol{V_{\widehat{n}}} - \frac{c}{\gamma-1} \\ \mathrm{c_v} \log(\frac{p}{\rho^\gamma}) \\ \boldsymbol{V_{\widehat{\tau}}} \\ \boldsymbol{V_{\widehat{n}}} + \frac{c}{\gamma-1} \end{pmatrix} \tag{A.22}$$

with $\boldsymbol{V_{\widehat{n}}} = \boldsymbol{V} \cdot \widehat{\boldsymbol{n}}$ and $\boldsymbol{V_{\widehat{\tau}}} = \boldsymbol{V} \cdot \widehat{\boldsymbol{\tau}}$. The unit vector normal $\widehat{\boldsymbol{n}}$ is oriented normal to the boundary facet and towards the flow orientation. Furthermore, in practice, $\mathrm{c_v}$ is assumed to be constant through the boundary and S is simplified by $s = \frac{p}{\rho^\gamma}$.

For an inlet, the Riemann Invariant at the boundary reads:

$$\boldsymbol{W_b^R} = \begin{pmatrix} (\mathcal{R}^-)_i \\ (s)_\infty \\ (\boldsymbol{V_{\widehat{\tau}}})_\infty \\ (\mathcal{R}^+)_\infty \end{pmatrix} \tag{A.23}$$

A.2.5.1 Boundary Riemann Invariant over primitive interior state Jacobian

The only dependance of the boundary Riemann Invariant with the primitive interior state is with the first Riemann Invariant $\boldsymbol{W_b^R}[0] = \boldsymbol{W_i^R}[0] = (\boldsymbol{V_{\widehat{n}}})_i - \frac{c_i}{\gamma_i - 1}$ with $c_i = \sqrt{\frac{\gamma_i p_i}{\rho_i}} = \sqrt{\frac{\gamma_i \boldsymbol{W_i^p}[4]}{\boldsymbol{W_i^p}[0]}}$.

The components of the Jacobian of the sound speed are:

$\frac{\partial c_i}{\partial \boldsymbol{W_i^p}[0]} = (\gamma_i p_i)^{\frac{1}{2}} (-\frac{1}{2}) \rho_i^{-\frac{3}{2}} = -\frac{1}{2}(\sqrt{\frac{\gamma_i p_i}{\rho_i}}) \rho_i^{-1} = -\frac{1}{2}\frac{c_i}{\rho_i} = -\frac{1}{2}\frac{c_i}{\boldsymbol{W_i^p}[0]}$

$\frac{\partial c_i}{\partial \boldsymbol{W_i^p}[4]} = (\frac{\gamma_i}{\rho_i})^{\frac{1}{2}} (\frac{1}{2}) p_i^{-\frac{1}{2}} = \frac{1}{2}\frac{c_i}{p_i} = \frac{1}{2}\frac{c_i}{\boldsymbol{W_i^p}[4]}$

Thus:

$$\frac{\partial c_i}{\partial \boldsymbol{W_i^p}} = (-\frac{1}{2}\frac{c_i}{\boldsymbol{W_i^p}[0]}, 0, 0, 0, \frac{1}{2}\frac{c_i}{\boldsymbol{W_i^p}[4]}) \tag{A.24}$$

On the other side:

$$\boldsymbol{V} \cdot \widehat{\boldsymbol{n}} = u\widehat{n}_x + v\widehat{n}_y + w\widehat{n}_z \tag{A.25}$$

That gives:

$$\frac{\partial \boldsymbol{V}_i \cdot \widehat{\boldsymbol{n}}}{\partial \boldsymbol{W_i^p}} = (0, \widehat{n}_x, \widehat{n}_y, \widehat{n}_z, 0) \tag{A.26}$$

The Jacobian $\frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}}$ becomes:

$$\frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}} = \begin{pmatrix} \frac{c_i}{(\gamma_i-1)\boldsymbol{W_i^p}[0]} & \widehat{n}_x & \widehat{n}_y & \widehat{n}_z & -\frac{c_i}{(\gamma_i-1)\boldsymbol{W_i^p}[4]} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A.27}$$

A.2.5.2 Boundary primitive state over boundary Riemann Invariant Jacobian

We remind the computation of the boundary state from the Riemann Invariant:

$$\boldsymbol{W_b^p} = \begin{pmatrix} (\frac{c_b^2}{\gamma_b \boldsymbol{W_b^R}[1]})^{\frac{1}{\gamma_b-1}} \\ \frac{1}{2}(\boldsymbol{W_b^R}[0] + \boldsymbol{W_b^R}[3])\widehat{\boldsymbol{n}} + \boldsymbol{W_b^R}[2]\widehat{\boldsymbol{\tau}} \\ \boldsymbol{W_b^R}[1]\boldsymbol{W_b^p}[0]^{\gamma_b} \end{pmatrix} \tag{A.28}$$

with $c_b = \frac{\gamma_b-1}{4}(\boldsymbol{W_b^R}[3] - \boldsymbol{W_b^R}[0])$.

The Jacobian of the sound speed is:

$$\frac{\partial c_b}{\partial \boldsymbol{W_b^R}[0]} = -\frac{\gamma_b - 1}{4} \tag{A.29}$$

$$\frac{\partial c_b^2}{\partial \boldsymbol{W_b^R}[0]} = -2\frac{\gamma_b - 1}{4}(\boldsymbol{W_b^R}[3] - \boldsymbol{W_b^R}[0]) = -c_b\frac{\gamma_b - 1}{2} \tag{A.30}$$

Details of the Jacobian computation:

$$\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{\gamma_b - 1}\left(\frac{1}{\gamma_b \boldsymbol{W_b^R}[1]}\right)^{\frac{1}{\gamma_b-1}} (c_b^2)^{\frac{2-\gamma_b}{\gamma_b-1}} \frac{\partial c_b^2}{\partial \boldsymbol{W_b^R}[0]} \tag{A.31}$$

$$= -\frac{1}{2}c_b\frac{1}{c_b^2}\left(\frac{c_b^2}{\gamma_b \boldsymbol{W_b^R}[1]}\right)^{\frac{1}{\gamma_b-1}} \tag{A.32}$$

$$= -\frac{\boldsymbol{W_b^p}[0]}{2c_b} \tag{A.33}$$

$$\frac{\partial \boldsymbol{W_b^p}[1]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{2}\widehat{n}_x \tag{A.34}$$

$$\frac{\partial \boldsymbol{W_b^p}[2]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{2}\widehat{n}_y \tag{A.35}$$

$$\frac{\partial \boldsymbol{W_b^p}[3]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{2}\widehat{n}_z \tag{A.36}$$

$$\begin{aligned}
\frac{\partial \boldsymbol{W_b^p}[4]}{\partial \boldsymbol{W_b^R}[0]} &= \boldsymbol{W_b^R}[1]\gamma_b \boldsymbol{W_b^p}[0]^{(\gamma_b-1)} \frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} && \text{(A.37)}\\
&= \gamma_b \frac{\boldsymbol{W_b^p}[4]}{\boldsymbol{W_b^p}[0]} \frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} && \text{(A.38)}\\
&= c_b^2 \frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} && \text{(A.39)}\\
&= -\frac{\boldsymbol{W_b^p}[0] c_b}{2} && \text{(A.40)}
\end{aligned}$$

Thus, the Jacobian $\frac{\partial c_b}{\partial \boldsymbol{W_b^R}[0]}$ becomes:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}} = \begin{pmatrix} -\frac{\boldsymbol{W_b^p}[0]}{2c_b} & X & X & X \\ \frac{1}{2}\widehat{n}_x & X & X & X \\ \frac{1}{2}\widehat{n}_y & X & X & X \\ \frac{1}{2}\widehat{n}_z & X & X & X \\ -\frac{\boldsymbol{W_b^p}[0]c_b}{2} & X & X & X \end{pmatrix} \tag{A.41}$$

The terms X of the Jacobian do not need to be computed as during the matrix multiplication $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}} \frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}}$, these terms are multiplied by 0 and disapear.

A.2.6 Riemann Invariant subsonic outlet

As for the Riemann Invariant subsonic inlet, the boundary state $\boldsymbol{W_b^p}$ is a function of the Riemann Invariant at the boundary $\boldsymbol{W_b^R}(\boldsymbol{W_i^p}, \boldsymbol{W_\infty^p})$ that depends on the interior state $\boldsymbol{W_i^p}$ and the freestream state $\boldsymbol{W_\infty^p}$:

$$\boldsymbol{W_b^p} = \boldsymbol{W_b^p}(\boldsymbol{W_b^R}(\boldsymbol{W_i^p}, \boldsymbol{W_\infty^p})) \tag{A.42}$$

The Jacobian $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}}$ is:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_i^p}} = \frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}} \frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}} \tag{A.43}$$

For an outlet, the Riemann Invariant at the boundary reads:

$$\boldsymbol{W_b^R} = \begin{pmatrix} (\mathcal{R}^-)_\infty \\ (s)_i \\ (\boldsymbol{V}_{\widehat{\boldsymbol{\tau}}})_i \\ (\mathcal{R}^+)_i \end{pmatrix} \tag{A.44}$$

The tangential velocity is a vector and is defined by: $(\boldsymbol{V}_{\widehat{\boldsymbol{\tau}}})_i = \boldsymbol{V}_i - 2(\boldsymbol{V}_i \cdot \widehat{\boldsymbol{n}})\widehat{\boldsymbol{n}}$.

A.2.6.1 Boundary Riemann Invariant over primitive interior state Jacobian

Needed variables:

$$\boldsymbol{W_b^R}[1] = \frac{\boldsymbol{W_i^p}[4]}{\boldsymbol{W_i^p}[0]^{\gamma_i}} \tag{A.45}$$

$$\boldsymbol{W_b^R}[5] = \boldsymbol{W_i^R}[5] = (\boldsymbol{V}_{\widehat{\boldsymbol{n}}})_i + \frac{c_i}{\gamma_i - 1} \tag{A.46}$$

with $c_i = \sqrt{\frac{\gamma_i p_i}{\rho_i}} = \sqrt{\frac{\gamma_i \boldsymbol{W_i^p}[4]}{\boldsymbol{W_i^p}[0]}}$.

Details of the Jacobian computation:

$$\forall k, \quad \frac{\partial \boldsymbol{W_b^R}[0]}{\partial \boldsymbol{W_i^p}[k]} = 0 \tag{A.47}$$

$$\frac{\partial \boldsymbol{W_b^R}[1]}{\partial \boldsymbol{W_i^p}[0]} = -\gamma_i \left(\frac{\boldsymbol{W_i^p}[4]}{\boldsymbol{W_i^p}[0]^{\gamma_i}} \right) \boldsymbol{W_i^p}[0]^{-1} \tag{A.48}$$

$$= -\frac{\gamma_i}{\boldsymbol{W_i^p}[0]} \boldsymbol{W_b^R}[1] \tag{A.49}$$

$$\forall k \in [1,3], \quad \frac{\partial \boldsymbol{W_b^R}[1]}{\partial \boldsymbol{W_i^p}[k]} = 0 \tag{A.50}$$

$$\frac{\partial \boldsymbol{W_b^R}[1]}{\partial \boldsymbol{W_i^p}[4]} = \frac{1}{\boldsymbol{W_i^p}[0]^{\gamma_i}} \tag{A.51}$$

$$\frac{\partial \boldsymbol{W_b^R}[2]}{\partial \boldsymbol{W_i^p}[0]} = 0 \tag{A.52}$$

$$\frac{\partial \boldsymbol{W_b^R}[2]}{\partial \boldsymbol{W_i^p}[1]} = \begin{pmatrix} 1 - \widehat{n}_x^2 \\ \widehat{n}_y \widehat{n}_x \\ \widehat{n}_z \widehat{n}_x \end{pmatrix} \tag{A.53}$$

$$\frac{\partial \boldsymbol{W_b^R}[2]}{\partial \boldsymbol{W_i^p}[2]} = \begin{pmatrix} \widehat{n}_x \widehat{n}_y \\ 1 - \widehat{n}_y^2 \\ \widehat{n}_z \widehat{n}_y \end{pmatrix} \tag{A.54}$$

$$\frac{\partial \boldsymbol{W_b^R}[2]}{\partial \boldsymbol{W_i^p}[3]} = \begin{pmatrix} \widehat{n}_x \widehat{n}_y \\ \widehat{n}_y \widehat{n}_z \\ 1 - \widehat{n}_z^2 \end{pmatrix} \tag{A.55}$$

$$\frac{\partial \boldsymbol{W_b^R}[2]}{\partial \boldsymbol{W_i^p}[4]} = 0 \tag{A.56}$$

$$\frac{\partial \boldsymbol{W_b^R}[4]}{\partial \boldsymbol{W_i^p}[0]} = -\frac{c_i}{(\gamma_i - 1)\boldsymbol{W_i^p}[0]} \tag{A.57}$$

$$\frac{\partial \boldsymbol{W_b^R}[4]}{\partial \boldsymbol{W_i^p}[1]} = \widehat{n}_x \tag{A.58}$$

$$\frac{\partial \boldsymbol{W_b^R}[4]}{\partial \boldsymbol{W_i^p}[2]} = \widehat{n}_y \tag{A.59}$$

$$\frac{\partial \boldsymbol{W_b^R}[4]}{\partial \boldsymbol{W_i^p}[3]} = \widehat{n}_z \tag{A.60}$$

$$\frac{\partial \boldsymbol{W_b^R}[4]}{\partial \boldsymbol{W_i^p}[4]} = \frac{c_i}{(\gamma_i - 1)\boldsymbol{W_i^p}[4]} \tag{A.61}$$

Thus, the Jacobian $\frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}}$ becomes:

$$\frac{\partial \boldsymbol{W_b^R}}{\partial \boldsymbol{W_i^p}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{\gamma_i}{\boldsymbol{W_i^p}[0]}\boldsymbol{W_b^R}[1] & 0 & 0 & 0 & \frac{1}{\boldsymbol{W_i^p}[0]^{\gamma_i}} \\ 0 & 1-\widehat{n}_x^2 & \widehat{n}_y\widehat{n}_x & \widehat{n}_z\widehat{n}_x & 0 \\ 0 & \widehat{n}_x\widehat{n}_y & 1-\widehat{n}_y^2 & \widehat{n}_z\widehat{n}_y & 0 \\ 0 & \widehat{n}_z\widehat{n}_x & \widehat{n}_z\widehat{n}_y & 1-\widehat{n}_z^2 & 0 \\ -\frac{c_i}{(\gamma_i-1)\boldsymbol{W_i^p}[0]} & \widehat{n}_x & \widehat{n}_y & \widehat{n}_z & \frac{c_i}{(\gamma_i-1)\boldsymbol{W_i^p}[4]} \end{pmatrix} \tag{A.62}$$

A.2.6.2 Boundary primitive state over Boundary Riemann Invariant Jacobian

Needed variables:

$$\boldsymbol{W_b^p} = \begin{pmatrix} (\frac{c_b^2}{\gamma_b \boldsymbol{W_b^R}[1]})^{\frac{1}{\gamma_b - 1}} \\ \frac{1}{2}(\boldsymbol{W_b^R}[0] + \boldsymbol{W_b^R}[3])\widehat{\boldsymbol{n}} + \boldsymbol{W_b^R}[2]\widehat{\boldsymbol{\tau}} \\ \boldsymbol{W_b^R}[1]\boldsymbol{W_b^p}[0]^{\gamma_b} \end{pmatrix} \tag{A.63}$$

with $c_b = \frac{\gamma_b - 1}{4}(\boldsymbol{W_b^R}[3] - \boldsymbol{W_b^R}[0])$.

Details of the Jacobian computation:

$$\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} = \left(\frac{1}{\gamma_b \boldsymbol{W_b^R}[1]}\right)^{\frac{1}{\gamma_b-1}} \frac{2}{\gamma_b-1} c_b^{\frac{1}{\gamma_b-1}-1} \frac{\partial c_b}{\partial \boldsymbol{W_b^p}[0]} \tag{A.64}$$

$$= \boldsymbol{W_b^p}[0] \frac{2}{(\gamma_b-1)c_b} \frac{\partial c_b}{\partial \boldsymbol{W_b^p}[0]} \tag{A.65}$$

$$= -\frac{\boldsymbol{W_b^p}[0]}{2c_b} \tag{A.66}$$

$$\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[1]} = \left(\frac{c_b^2}{\gamma_b}\right)^{\frac{1}{\gamma_b-1}} \frac{-1}{\gamma_b-1} \boldsymbol{W_b^R}[1]^{(-\frac{1}{\gamma_b-1}-1)} \tag{A.67}$$

$$= -\frac{\boldsymbol{W_b^p}[0]}{(\gamma_b-1)\boldsymbol{W_b^R}[1]} \tag{A.68}$$

$$\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[2]} = 0 \tag{A.69}$$

$$\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[3]} = \boldsymbol{W_b^p}[0] \frac{2}{\gamma_b-1} \frac{\partial c_b}{\partial \boldsymbol{W_b^R}[3]} \tag{A.70}$$

$$= \frac{\boldsymbol{W_b^p}[0]}{2c_b} \tag{A.71}$$

$$\frac{\partial \boldsymbol{W_b^p}[1-2-3]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{2}\widehat{\boldsymbol{n}} \tag{A.72}$$

$$\frac{\partial \boldsymbol{W_b^p}[1-2-3]}{\partial \boldsymbol{W_b^R}[1]} = 0 \tag{A.73}$$

$$\frac{\partial \boldsymbol{W_b^p}[1-2-3]}{\partial \boldsymbol{W_b^R}[2]} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \tag{A.74}$$

$$\frac{\partial \boldsymbol{W_b^p}[1-2-3]}{\partial \boldsymbol{W_b^R}[3]} = \frac{1}{2}\widehat{\boldsymbol{n}} \tag{A.75}$$

$$\frac{\partial \boldsymbol{W_b^p}[4]}{\partial \boldsymbol{W_b^R}[0]} = \frac{1}{\gamma_b}\left(c_b^2 \frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[0]} + \boldsymbol{W_b^p}[0] 2c_b \frac{\partial c_b}{\partial \boldsymbol{W_b^R}[0]}\right) \tag{A.76}$$

$$= -\frac{\boldsymbol{W_b^p}[0]c_b}{2} \tag{A.77}$$

$$\frac{\partial \boldsymbol{W_b^p}[4]}{\partial \boldsymbol{W_b^R}[1]} = \frac{c_b^2}{\gamma_b}\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[1]} \tag{A.78}$$

$$= \frac{c_b^2}{\gamma_b}\left(-\frac{\boldsymbol{W_b^p}[0]}{(\gamma_b-1)\boldsymbol{W_b^R}[1]}\right) \tag{A.79}$$

$$= \frac{\gamma_b \boldsymbol{W_b^p}[4]}{\boldsymbol{W_b^p}[0]\gamma_b}\left(-\frac{\boldsymbol{W_b^p}[0]}{(\gamma_b-1)\frac{\boldsymbol{W_b^p}[4]}{\boldsymbol{W_b^p}[0]^{\gamma_b}}}\right) \tag{A.80}$$

$$= -\frac{\boldsymbol{W_b^p}[0]^{\gamma_b}}{\gamma_b-1} \tag{A.81}$$

$$\frac{\partial \boldsymbol{W_b^p}[4]}{\partial \boldsymbol{W_b^R}[2]} = 0 \tag{A.82}$$

$$\frac{\partial \boldsymbol{W_b^p}[4]}{\partial \boldsymbol{W_b^R}[3]} = \frac{c_b^2}{\gamma_b}\frac{\partial \boldsymbol{W_b^p}[0]}{\partial \boldsymbol{W_b^R}[3]} + \frac{\boldsymbol{W_b^p}[0]}{\gamma_b}\partial_{\boldsymbol{W_b^R}[3]}c_b^2 \tag{A.83}$$

$$= \frac{c_b \boldsymbol{W_b^p}[0]}{2} \tag{A.84}$$

Thus, the Jacobian $\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}}$ becomes:

$$\frac{\partial \boldsymbol{W_b^p}}{\partial \boldsymbol{W_b^R}} = \begin{pmatrix} -\frac{\boldsymbol{W_b^p}[0]}{2c_b} & -\frac{\boldsymbol{W_b^p}[0]}{(\gamma_b-1)\boldsymbol{W_b^R}[1]} & 0 & 0 & 0 & \frac{\boldsymbol{W_b^p}[0]}{2c_b} \\ \frac{1}{2}\widehat{n}_x & 0 & 1 & 0 & 0 & \frac{1}{2}\widehat{n}_x \\ \frac{1}{2}\widehat{n}_y & 0 & 0 & 1 & 0 & \frac{1}{2}\widehat{n}_y \\ \frac{1}{2}\widehat{n}_z & 0 & 0 & 0 & 1 & \frac{1}{2}\widehat{n}_z \\ -\frac{\boldsymbol{W_b^p}[0]c_b}{2} & -\frac{\boldsymbol{W_b^p}[0]^{\gamma_b}}{\gamma_b-1} & 0 & 0 & 0 & \frac{c_b\boldsymbol{W_b^p}[0]}{2} \end{pmatrix} \tag{A.85}$$

A.3 Viscous Jacobians

A.3.1 General problem of finding the Jacobian for a numerical viscous flux

A numerical viscous flux can be expressed as [Fezoui 1989]:

$$\boldsymbol{\Upsilon_{T,i}} = \boldsymbol{\Upsilon}((\boldsymbol{W_0^c}, \boldsymbol{W_1^c}, ..., \boldsymbol{W_k^c}), \boldsymbol{\tau}(\mu, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}), \boldsymbol{q}(\gamma, \kappa(\mu, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon), \boldsymbol{\eta_{,i,T}}) \tag{A.86}$$

by introducing the element gradient with a P1-Approximation for any variable:

$$\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c} = \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}((\boldsymbol{W_0^c}, \boldsymbol{W_1^c}, ..., \boldsymbol{W_k^c}), (\boldsymbol{\nabla}\varphi_0^T, \boldsymbol{\nabla}\varphi_1^T, ..., \boldsymbol{\nabla}\varphi_k^T), (\alpha_{T\cap C_0}, \alpha_{T\cap C_1}, ..., \alpha_{T\cap C_k})) \tag{A.87}$$

the shear stress tensor $\boldsymbol{\tau}$ is:

$$\boldsymbol{\tau}(\mu, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) = 2\mu\mathbf{D} - \frac{1}{3}(\mathrm{tr}(2\mu\mathbf{D}))\boldsymbol{I_n} \tag{A.88}$$

and the heat flux $\boldsymbol{q}$ by:

$$\begin{aligned}\boldsymbol{q}(\gamma, \kappa(\mu, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon) &= -\gamma\kappa\boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon \\ &= -\gamma\frac{\mu}{\mathrm{Pr}}\boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon\end{aligned} \tag{A.89}$$

The Jacobian of the numerical viscous flux is expressed as following:

$$\frac{\partial\boldsymbol{\Upsilon_{T,i}}}{\partial\boldsymbol{W_j^c}} = \frac{\partial\boldsymbol{\Upsilon}((\boldsymbol{W_0^c}, \boldsymbol{W_1^c}, ..., \boldsymbol{W_k^c}), \boldsymbol{\tau}(\mu, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}), \boldsymbol{q}(\gamma, \kappa(\mu, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon), \boldsymbol{\eta_{,i,T}})}{\partial\boldsymbol{W_j^c}} \tag{A.90}$$

A.3.2 Interior numerical viscous flux Jacobian computation

The numerical viscous flux on a internal tetrahedron T can be expressed as:

$$\frac{\partial\boldsymbol{\Upsilon_{T,i}}}{\partial\boldsymbol{W_j^c}} = \frac{\partial\boldsymbol{\Upsilon}(\overline{\boldsymbol{W^c}}^w_{|tetra}, \boldsymbol{\tau}(\mu, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}), \boldsymbol{q}(\gamma, \kappa(\mu, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}}\varepsilon), \boldsymbol{\eta_{,i,T}})}{\partial\boldsymbol{W_j^c}} \tag{A.91}$$

with $\overline{\boldsymbol{W^c}}^w_{|T}$ the weighted average on the tetrahedron:

$$\begin{aligned}\overline{\boldsymbol{W^c}}^w_{|T} &= \frac{\sum_{k\in T}\alpha_{T\cap C_k}\boldsymbol{W_k^c}}{\sum_{k\in T}\alpha_{T\cap C_k}} \\ &= \sum_{k\in T}\alpha_{T\cap C_k}\boldsymbol{W_k^c}\end{aligned} \tag{A.92}$$

as $\sum_{k\in T}\alpha_{T\cap C_k} = 1$.

For a regular cell vertex centered formulation in 2D, the weighted average becomes (i.e. with no use of Barth cell or similar approach):

$$\overline{\boldsymbol{W^c}}^w_{|T} = \frac{1}{3}\sum_{k\in T}\boldsymbol{W_k^c} \tag{A.93}$$

This is the arithmetic mean.

Shear stress tensor Jacobian: The term $\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c})$ is expressed as:

$$\boldsymbol{\tau}_{xx} = \frac{2}{3}\overline{\mu}^m_{|T}(2(\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[1])[0] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[2])[1] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[3])[2]) \tag{A.94}$$

$$\boldsymbol{\tau}_{yy} = \frac{2}{3}\overline{\mu}^m_{|T}(2(\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[2])[1] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[3])[2] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[1])[0]) \tag{A.95}$$

$$\boldsymbol{\tau}_{zz} = \frac{2}{3}\overline{\mu}^m_{|T}(2(\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[3])[2] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[1])[0] - (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[2])[1]) \tag{A.96}$$

$$\boldsymbol{\tau}_{xy} = \overline{\mu}^m_{|T}((\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[1])[1] + (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[2])[0]) \tag{A.97}$$

$$\boldsymbol{\tau}_{yz} = \overline{\mu}^m_{|T}((\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[2])[2] + (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[3])[1]) \tag{A.98}$$

$$\boldsymbol{\tau}_{zx} = \overline{\mu}^m_{|T}((\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[3])[0] + (\boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^p}[1])[2]) \tag{A.99}$$

The shear stress tensor $\boldsymbol{\tau}$ is symmetric.

With the P1 element approximation, the shear stress tensor becomes:

$$\boldsymbol{\tau}_{xx} = \frac{2}{3}\overline{\mu}^m_{|T}\sum_{k\in T}(2\boldsymbol{W^p_k}[1]\partial_x\varphi^T_k - \boldsymbol{W^p_k}[2]\partial_y\varphi^T_k - \boldsymbol{W^p_k}[3]\partial_z\varphi^T_k) \tag{A.100}$$

$$\boldsymbol{\tau}_{yy} = \frac{2}{3}\overline{\mu}^m_{|T}\sum_{k\in T}(2\boldsymbol{W^p_k}[2]\partial_y\varphi^T_k - \boldsymbol{W^p_k}[3]\partial_z\varphi^T_k - \boldsymbol{W^p_k}[1]\partial_x\varphi^T_k) \tag{A.101}$$

$$\boldsymbol{\tau}_{zz} = \frac{2}{3}\overline{\mu}^m_{|T}\sum_{k\in T}(2\boldsymbol{W^p_k}[3]\partial_z\varphi^T_k - \boldsymbol{W^p_k}[1]\partial_x\varphi^T_k - \boldsymbol{W^p_k}[2]\partial_y\varphi^T_k) \tag{A.102}$$

$$\boldsymbol{\tau}_{xy} = \overline{\mu}^m_{|T}\sum_{k\in T}(\boldsymbol{W^p_k}[1]\partial_y\varphi^T_k + \boldsymbol{W^p_k}[2]\partial_x\varphi^T_k) \tag{A.103}$$

$$\boldsymbol{\tau}_{yz} = \overline{\mu}^m_{|T}\sum_{k\in T}(\boldsymbol{W^p_k}[2]\partial_z\varphi^T_k + \boldsymbol{W^p_k}[3]\partial_y\varphi^T_k) \tag{A.104}$$

$$\boldsymbol{\tau}_{zx} = \overline{\mu}^m_{|T}\sum_{k\in T}(\boldsymbol{W^p_k}[3]\partial_x\varphi^T_k + \boldsymbol{W^p_k}[1]\partial_z\varphi^T_k) \tag{A.105}$$

Thus, the Jacobian of the shear stress tensor is (for any $i \in T$):

$$\frac{\partial\boldsymbol{\tau}_{xx}}{\partial\boldsymbol{W^c_i}[0]} = -\frac{2}{3}\overline{\mu}^m_{|T}\frac{1}{\boldsymbol{W^p_i}[0]}(2\boldsymbol{W^p_i}[1]\partial_x\varphi^T_i - \boldsymbol{W^p_i}[2]\partial_y\varphi^T_i - \boldsymbol{W^p_i}[3]\partial_z\varphi^T_i) \tag{A.106}$$

$$\frac{\partial\boldsymbol{\tau}_{xx}}{\partial\boldsymbol{W^c_i}[1]} = \frac{4}{3}\overline{\mu}^m_{|T}\frac{1}{\boldsymbol{W^p_i}[0]}\partial_x\varphi^T_i \tag{A.107}$$

$$\frac{\partial\boldsymbol{\tau}_{xx}}{\partial\boldsymbol{W^c_i}[2]} = -\frac{2}{3}\overline{\mu}^m_{|T}\frac{1}{\boldsymbol{W^p_i}[0]}\partial_y\varphi^T_i \tag{A.108}$$

$$\frac{\partial\boldsymbol{\tau}_{xx}}{\partial\boldsymbol{W^c_i}[3]} = -\frac{2}{3}\overline{\mu}^m_{|T}\frac{1}{\boldsymbol{W^p_i}[0]}\partial_z\varphi^T_i \tag{A.109}$$

$$\frac{\partial \boldsymbol{\tau}_{xx}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[4]} = 0 \tag{A.110}$$

$$\frac{\partial \boldsymbol{\tau}_{yy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[0]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(2\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[2]\partial_y\varphi_i^T - \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[3]\partial_z\varphi_i^T - \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[1]\partial_x\varphi_i^T) \tag{A.111}$$

$$\frac{\partial \boldsymbol{\tau}_{yy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[1]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_x\varphi_i^T \tag{A.112}$$

$$\frac{\partial \boldsymbol{\tau}_{yy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[2]} = \frac{4}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_y\varphi_i^T \tag{A.113}$$

$$\frac{\partial \boldsymbol{\tau}_{yy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[3]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_z\varphi_i^T \tag{A.114}$$

$$\frac{\partial \boldsymbol{\tau}_{yy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[4]} = 0 \tag{A.115}$$

$$\frac{\partial \boldsymbol{\tau}_{zz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[0]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(2\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[3]\partial_z\varphi_i^T - \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[1]\partial_x\varphi_i^T - \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[2]\partial_y\varphi_i^T) \tag{A.116}$$

$$\frac{\partial \boldsymbol{\tau}_{zz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[1]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_x\varphi_i^T \tag{A.117}$$

$$\frac{\partial \boldsymbol{\tau}_{zz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[2]} = -\frac{2}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_y\varphi_i^T \tag{A.118}$$

$$\frac{\partial \boldsymbol{\tau}_{zz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[3]} = \frac{4}{3}\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}\partial_z\varphi_i^T \tag{A.119}$$

$$\frac{\partial \boldsymbol{\tau}_{zz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[4]} = 0 \tag{A.120}$$

$$\frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[0]} = -\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[1]\partial_y\varphi_i^T + \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[2]\partial_x\varphi_i^T) \tag{A.121}$$

$$\frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[1]} = \overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(\partial_y\varphi_i^T) \tag{A.122}$$

$$\frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[2]} = \overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(\partial_x\varphi_i^T) \tag{A.123}$$

$$\frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[3]} = 0 \tag{A.124}$$

$$\frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[4]} = 0 \tag{A.125}$$

$$\frac{\partial \boldsymbol{\tau}_{yz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[0]} = -\overline{\mu}_{|T}^{m}\frac{1}{\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[0]}(\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[2]\partial_z\varphi_i^T + \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{p}}[3]\partial_y\varphi_i^T) \tag{A.126}$$

$$\frac{\partial \boldsymbol{\tau}_{yz}}{\partial \boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}[1]} = 0 \tag{A.127}$$

$$\frac{\partial \boldsymbol{\tau}_{yz}}{\partial \boldsymbol{W_i^c}[2]} = \overline{\mu}^m_{|T} \frac{1}{\boldsymbol{W_i^p}[0]} (\partial_z \varphi_i^T) \tag{A.128}$$

$$\frac{\partial \boldsymbol{\tau}_{yz}}{\partial \boldsymbol{W_i^c}[3]} = \overline{\mu}^m_{|T} \frac{1}{\boldsymbol{W_i^p}[0]} (\partial_y \varphi_i^T) \tag{A.129}$$

$$\frac{\partial \boldsymbol{\tau}_{yz}}{\partial \boldsymbol{W_i^c}[4]} = 0 \tag{A.130}$$

$$\frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[0]} = -\overline{\mu}^m_{|T} \frac{1}{\boldsymbol{W_i^p}[0]} (\boldsymbol{W_i^p}[1] \partial_z \varphi_i^T + \boldsymbol{W_i^p}[3] \partial_x \varphi_i^T) \tag{A.131}$$

$$\frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[1]} = \overline{\mu}^m_{|T} \frac{1}{\boldsymbol{W_i^p}[0]} (\partial_z \varphi_i^T) \tag{A.132}$$

$$\frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[2]} = 0 \tag{A.133}$$

$$\frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[3]} = \overline{\mu}^m_{|T} \frac{1}{\boldsymbol{W_i^p}[0]} (\partial_x \varphi_i^T) \tag{A.134}$$

$$\frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[4]} = 0 \tag{A.135}$$

Heat flux Jacobian: The heat flux is:

$$\boldsymbol{q}(\gamma, \kappa(\overline{\mu}^m_{|T}, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}} \varepsilon) = -\gamma \frac{\overline{\mu}^m_{|T}}{\mathrm{Pr}} \boldsymbol{\nabla}_{|\boldsymbol{T}} \varepsilon \tag{A.136}$$

with

$$\varepsilon = \frac{\boldsymbol{W^p}[4]}{(\gamma - 1)\boldsymbol{W^p}[0]} = \frac{1}{\boldsymbol{W^c}[0]} \left(\boldsymbol{W^c}[4] - \frac{\boldsymbol{W^c}[1]^2 + \boldsymbol{W^c}[2]^2 + \boldsymbol{W^c}[3]^2}{2\boldsymbol{W^c}[0]} \right) \tag{A.137}$$

The variables $\overline{\mu}^m_{|T}$, Pr and γ are locally frozen and thus are supposed to not depend on primitive variables $\boldsymbol{W^p}$ (or conservative ones).

With the P1-Approximation:

$$\boldsymbol{q}(\gamma, \kappa(\overline{\mu}^m_{|T}, \mathrm{Pr}), \boldsymbol{\nabla}_{|\boldsymbol{T}} \varepsilon) = -\theta \sum_{k \in T} \varepsilon_k \boldsymbol{\nabla} \varphi_k^T \tag{A.138}$$

with $\theta = \gamma \frac{\overline{\mu}^m_{|T}}{\mathrm{Pr}}$.

Thus, the Jacobian is:

$$\begin{aligned} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{W_i^c}[0]} &= -\theta \frac{\partial \varepsilon_i}{\partial \boldsymbol{W_i^c}[0]} \boldsymbol{\nabla} \varphi_i^T \\ &= \theta \frac{1}{\boldsymbol{W_i^c}[0]^2} \left(\boldsymbol{W_i^c}[4] - \frac{\boldsymbol{W_i^c}[1]^2 + \boldsymbol{W_i^c}[2]^2 + \boldsymbol{W_i^c}[3]^2}{\boldsymbol{W_i^c}[0]} \right) \boldsymbol{\nabla} \varphi_i^T \\ &= \theta \frac{1}{\boldsymbol{W_i^p}[0]} \left(E - \boldsymbol{W_i^p}[1]^2 + \boldsymbol{W_i^p}[2]^2 + \boldsymbol{W_i^p}[3]^2 \right) \boldsymbol{\nabla} \varphi_i^T \end{aligned} \tag{A.139}$$

with $E = \frac{\boldsymbol{W_i^p}[4]}{(\gamma-1)\boldsymbol{W_i^p}[0]} + \frac{1}{2}(\boldsymbol{W_i^p}[1]^2 + \boldsymbol{W_i^p}[2]^2 + \boldsymbol{W_i^p}[3]^2)$

$$\begin{aligned}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{W_i^c}[1]} &= -\theta \frac{\partial \varepsilon_i}{\partial \boldsymbol{W_i^c}[1]} \boldsymbol{\nabla} \varphi_i^T \\ &= \theta \frac{1}{\boldsymbol{W_i^p}[0]} \boldsymbol{W_i^p}[1] \boldsymbol{\nabla} \varphi_i^T\end{aligned} \tag{A.140}$$

$$\begin{aligned}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{W_i^c}[2]} &= -\theta \frac{\partial \varepsilon_i}{\partial \boldsymbol{W_i^c}[2]} \boldsymbol{\nabla} \varphi_i^T \\ &= \theta \frac{1}{\boldsymbol{W_i^p}[0]} \boldsymbol{W_i^p}[2] \boldsymbol{\nabla} \varphi_i^T\end{aligned} \tag{A.141}$$

$$\begin{aligned}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{W_i^c}[3]} &= -\theta \frac{\partial \varepsilon_i}{\partial \boldsymbol{W_i^c}[3]} \boldsymbol{\nabla} \varphi_i^T \\ &= \theta \frac{1}{\boldsymbol{W_i^p}[0]} \boldsymbol{W_i^p}[3] \boldsymbol{\nabla} \varphi_i^T\end{aligned} \tag{A.142}$$

$$\begin{aligned}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{W_i^c}[4]} &= -\theta \frac{\partial \varepsilon_i}{\partial \boldsymbol{W_i^c}[4]} \boldsymbol{\nabla} \varphi_i^T \\ &= -\theta \frac{1}{\boldsymbol{W_i^p}[0]} \boldsymbol{\nabla} \varphi_i^T\end{aligned} \tag{A.143}$$

Full Navier-Stokes Jacobian To compute the full internal Jacobian, the only missing term is the energy term introduced by the shear stress tensor energy:

$$\boldsymbol{\tau}(\overline{\mu}_{|T}^m, \boldsymbol{\nabla}_{|\boldsymbol{T}} \boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}_{|T}^m = \begin{pmatrix} \boldsymbol{\tau}_{xx}\overline{u}_{|T}^m + \boldsymbol{\tau}_{xy}\overline{v}_{|T}^m + \boldsymbol{\tau}_{xz}\overline{w}_{|T}^m \\ \boldsymbol{\tau}_{yx}\overline{u}_{|T}^m + \boldsymbol{\tau}_{yy}\overline{v}_{|T}^m + \boldsymbol{\tau}_{yz}\overline{w}_{|T}^m \\ \boldsymbol{\tau}_{zx}\overline{u}_{|T}^m + \boldsymbol{\tau}_{zy}\overline{v}_{|T}^m + \boldsymbol{\tau}_{zz}\overline{w}_{|T}^m \end{pmatrix} \tag{A.144}$$

with

$$\overline{\boldsymbol{V}}_{|T}^m = \frac{1}{N_s \in T} \sum_{k \in T} \boldsymbol{V}_k \tag{A.145}$$

with $N_s \in T$ the number of vertices in the tetra T.

The Jacobian of this term is expressed as following:

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}_{|T}^m, \boldsymbol{\nabla}_{|\boldsymbol{T}} \boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}_{|T}^m)}{\partial \boldsymbol{W_i^c}[0]} = \frac{\partial \boldsymbol{\tau}(\overline{\mu}_{|T}^m, \boldsymbol{\nabla}_{|\boldsymbol{T}} \boldsymbol{W^c})}{\partial \boldsymbol{W_i^c}[0]} \cdot \overline{\boldsymbol{V}}_{|T}^m + \boldsymbol{\tau}(\overline{\mu}_{|T}^m, \boldsymbol{\nabla}_{|\boldsymbol{T}} \boldsymbol{W^c}) \cdot \frac{\partial \overline{\boldsymbol{V}}_{|T}^m}{\partial \boldsymbol{W_i^c}[0]} \tag{A.146}$$

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}^m_{|T})[0]}{\partial \boldsymbol{W_i^c}[0]} = \frac{\partial \boldsymbol{\tau}_{xx}}{\partial \boldsymbol{W_i^c}[0]}\overline{u}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W_i^c}[0]}\overline{v}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[0]}\overline{w}^m_{|T} - \frac{1}{N_s \in T}\frac{1}{\boldsymbol{W_i^p}[0]}(\boldsymbol{\tau}_{xx}\boldsymbol{W_i^p}[1] + \boldsymbol{\tau}_{xy}\boldsymbol{W_i^p}[2] + \boldsymbol{\tau}_{xz}\boldsymbol{W_i^p}[3]) \tag{A.147}$$

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}^m_{|T})[0]}{\partial \boldsymbol{W_i^c}[1]} = \frac{\partial \boldsymbol{\tau}_{xx}}{\partial \boldsymbol{W_i^c}[1]}\overline{u}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W_i^c}[1]}\overline{v}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[1]}\overline{w}^m_{|T} + \frac{1}{N_s \in T}\frac{1}{\boldsymbol{W_i^p}[0]}(\boldsymbol{\tau}_{xx}\boldsymbol{W_i^p}[1]) \tag{A.148}$$

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}^m_{|T})[0]}{\partial \boldsymbol{W_i^c}[2]} = \frac{\partial \boldsymbol{\tau}_{xx}}{\partial \boldsymbol{W_i^c}[2]}\overline{u}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W_i^c}[2]}\overline{v}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[2]}\overline{w}^m_{|T} + \frac{1}{N_s \in T}\frac{1}{\boldsymbol{W_i^p}[0]}(\boldsymbol{\tau}_{xy}\boldsymbol{W_i^p}[2]) \tag{A.149}$$

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}^m_{|T})[0]}{\partial \boldsymbol{W_i^c}[3]} = \frac{\partial \boldsymbol{\tau}_{xx}}{\partial \boldsymbol{W_i^c}[3]}\overline{u}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xy}}{\partial \boldsymbol{W_i^c}[3]}\overline{v}^m_{|T} + \frac{\partial \boldsymbol{\tau}_{xz}}{\partial \boldsymbol{W_i^c}[3]}\overline{w}^m_{|T} + \frac{1}{N_s \in T}\frac{1}{\boldsymbol{W_i^p}[0]}(\boldsymbol{\tau}_{xz}\boldsymbol{W_i^p}[3]) \tag{A.150}$$

$$\frac{\partial(\boldsymbol{\tau}(\overline{\mu}^m_{|T}, \boldsymbol{\nabla}_{|\boldsymbol{T}}\boldsymbol{W^c}) \cdot \overline{\boldsymbol{V}}^m_{|T})[0]}{\partial \boldsymbol{W_i^c}[4]} = 0 \tag{A.151}$$

with $\overline{u}^m_{|T} = \frac{1}{N_s \in T}\sum_{k \in T}\boldsymbol{W_i^p}[1]$, $\overline{v}^m_{|T} = \frac{1}{N_s \in T}\sum_{k \in T}\boldsymbol{W_i^p}[2]$ and $\overline{w}^m_{|T} = \frac{1}{N_s \in T}\sum_{k \in T}\boldsymbol{W_i^p}[3]$.

Bibliography

[Amitay 2001] Michael Amitay, Douglas R Smith, Valdis Kibens, David E Parekh and Ari Glezer. *Aerodynamic flow control over an unconventional airfoil using synthetic jet actuators.* AIAA journal, vol. 39, no. 3, pages 361–370, 2001. (Cited on page 3.)

[Ashcroft 2004] G. Ashcroft and J. Schulz. *Numerical modelling of wake-jet interaction with application to active noise control in turbomachinery.* AIAA journal, 2004. (Cited on page 45.)

[Barth 1995] T.J. Barth. *Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations.* In Lecture Notes Presented at the VKI Lecture Series, volume 1994, 1995. (Cited on page 34.)

[Batten 1997a] P. Batten, N. Clarke, C. Lambert and DM Causon. *On the choice of wavespeeds for the HLLC Riemann solver.* SIAM Journal on Scientific Computing, vol. 18, page 1553, 1997. (Cited on page 39.)

[Batten 1997b] P Batten, MA Leschziner and UC Goldberg. *Average-state Jacobians and implicit methods for compressible viscous and turbulent flows.* Journal of computational physics, vol. 137, no. 1, pages 38–78, 1997. (Cited on page 188.)

[Belme 2011] Anca Belme. *Aérodynamique instationnaire et méthode adjointe.* PhD thesis, Université Nice Sophia Antipolis, 2011. (Cited on pages 5 and 91.)

[Beyer 2002] Hans-Georg Beyer and Hans-Paul Schwefel. *Evolution strategies–A comprehensive introduction.* Natural computing, vol. 1, no. 1, pages 3–52, 2002. (Cited on pages 104 and 105.)

[Bonnans 2006] Joseph-Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal and Claudia A Sagastizábal. Numerical optimization: theoretical and practical aspects. Springer, 2006. (Cited on page 91.)

[Brezillon 2009] Joël Brezillon and Richard P Dwight. *Aerodynamic Shape Optimization Using the Discrete Adjoint of the Navier-Stokes Equations: Applications towards Complex 3D Configurations.* In KATnet II Conference on Key Aerodynamic Technologies paper, 2009. (Cited on page 91.)

[Buche 2005] D. Buche, N.N. Schraudolph and P. Koumoutsakos. *Accelerating evolutionary algorithms with gaussian process fitness function models.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 35, no. 2, pages 183–194, 2005. (Cited on page 92.)

[Burman 2001] Jörgen Burman and B Rikard Gebart. *Influence from numerical noise in the objective function for flow design optimisation.* International Journal of

Numerical Methods for Heat & Fluid Flow, vol. 11, no. 1, pages 6–19, 2001. (Cited on page 178.)

[Camarri 2004] Simone Camarri, Maria Vittoria Salvetti, B Koobus and Alain Dervieux. *A low-diffusion MUSCL scheme for LES on unstructured grids.* Computers & fluids, vol. 33, no. 9, pages 1101–1129, 2004. (Cited on page 35.)

[Capizzano 2005] Francesco Capizzano, Pietro Catalano, Claudio Marongiu and Pier Luigi Vitagliano. *U-RANS modelling of turbulent flows controlled by synthetic jets.* In 35th AIAA Fluid Dynamics Conference and Exhibit, Toronto, Canada, AIAA paper, volume 5015, 2005. (Cited on page 27.)

[Carpy 2006] S. Carpy and R. Manceau. *Turbulence modelling of statistically periodic flows: Synthetic jet into quiescent air.* International Journal of Heat and Fluid Flow, vol. 27, no. 5, pages 756 – 767, 2006. <ce:title>Special issue of the 6th International Symposium on Engineering Turbulence Modelling and Measurements â“ ETMM6</ce:title>. (Cited on page 24.)

[Coles 1968] DE Coles and EA Hirst. *Computation of turbulent boundary layers.* In Proceedings of the AFOSR-IFP Stanford Conference on Turbulent Boundary-Layer Prediction, volume 2, 1968. (Cited on page 50.)

[Commission 2011] European Commission*et al.* *Flightpath 2050. Europe's Vision for Aviation.* Report of the High Level Group on Aviation Research, Publications Office of the European Union, Luxembourg, 2011. (Cited on page 3.)

[Dandois 2006] Julien Dandois, Eric Garnier and Pierre Sagaut. *Unsteady simulation of synthetic jet in a crossflow.* AIAA journal, vol. 44, no. 2, pages 225–238, 2006. (Cited on page 25.)

[Dandois 2007] Julien Dandois. *Contrôle des décollements par jet synthétique.* PhD thesis, Paris 6, 2007. (Cited on page 26.)

[Debiez 1998] Christophe Debiez, Alain Dervieux, Katherine Mer and Boniface Nkonga. *Computation of unsteady flows with mixed finite volume/finite element upwind methods.* International Journal for Numerical Methods in Fluids, vol. 27, no. 1-4, pages 193–206, 1998. (Cited on page 36.)

[Debiez 2000] Christophe Debiez and Alain Dervieux. *Mixed-element-volume MUSCL methods with weak viscosity for steady and unsteady flow calculations.* Computers & fluids, vol. 29, no. 1, pages 89–118, 2000. (Cited on page 33.)

[Deng 1999] GB Deng and M. Visonneau. *Comparison of explicit algebraic stress models and second-order turbulence closures for steady flows around ships.* In Proc. 7th Int. Conf. on Numerical Ship Hydrodynamics, Nantes, France, 1999. (Cited on page 24.)

[Dervieux 1985] A Dervieux. *Steady Euler simulations using unstructured meshes.* In In Von Karman Inst. for Fluid Dynamics Computational Fluid Dynamics, Vol. 1 81 p (SEE N86-18652 09-34), volume 1, 1985. (Cited on page 33.)

[Desideri 1988] JA Desideri and A Dervieux. *Compressible flow solvers using unstructured grids.* VKI, Computational Fluid Dynamics,, vol. 2, 1988. (Cited on page 33.)

[Diggle 2007] Peter Diggle and Paulo Justiniano Ribeiro. Model-based geostatistics. Springer, 2007. (Cited on page 98.)

[Donovan 1998] John F Donovan, Linda D Kral and Andrew W Cary. *Active flow control applied to an airfoil.* AIAA paper, vol. 210, 1998. (Cited on pages 5 and 153.)

[Driver 1985] D.M. Driver and H.L. Seegmiller. *Features of a reattaching turbulent shear layer in divergent channel flow.* AIAA journal, vol. 23, no. 2, pages 163–171, 1985. (Cited on page 156.)

[Duchaine 2009] F. Duchaine, T. Morel and L.Y.M. Gicquel. *Computational-Fluid-Dynamics-Based Kriging Optimization Tool for Aeronautical Combustion Chambers.* AIAA Journal, vol. 47, no. 3, pages 631–645, 2009. (Cited on page 93.)

[Duvigneau 2003] R. Duvigneau, M. Visonneau and G.B. Deng. *On the role played by turbulence closures in hull shape optimization at model and full scale.* Journal of marine science and technology, vol. 8, no. 1, pages 11–25, 2003. (Cited on page 156.)

[Duvigneau 2006] Régis Duvigneau and Michel Visonneau. *Optimization of a synthetic jet actuator for aerodynamic stall control.* Computers & fluids, vol. 35, no. 6, pages 624–638, 2006. (Cited on pages 126 and 153.)

[Duvigneau 2012] Régis Duvigneau and Praveen Chandrashekar. *Kriging-based optimization applied to flow control.* International Journal for Numerical Methods in Fluids, vol. 69, no. 11, pages 1701–1714, August 2012. (Cited on pages 92 and 93.)

[Dwight 2006] R.P. Dwight. *Efficiency improvements of RANS-based analysis and optimization using implicit and adjoint methods on unstructured grids.* PhD thesis, the University of Manchester, 2006. (Cited on pages 42 and 91.)

[Eberhart 1995] Russ C Eberhart and James Kennedy. *A new optimizer using particle swarm theory.* In Proceedings of the sixth international symposium on micro machine and human science, volume 1, pages 39–43. New York, NY, 1995. (Cited on page 5.)

[Ekaterinaris 2003] John A Ekaterinaris. *Active flow control of wing separated flow.* In ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference, pages 2727–2735. American Society of Mechanical Engineers, 2003. (Cited on pages 4 and 5.)

[Fezoui 1989] Loula Fezoui Fatima, Stephane Lanteri, Bernard Larrouturou and Christian Olivier. *Resolution numerique des equations de Navier-Stokes pour un fluide compressible en maillage triangulaire.* Rapport de recherche RR-1033, INRIA, 1989. (Cited on pages 33, 36, 42 and 199.)

[Francescatto 1998] J. Francescatto. *Méthodes multigrilles par agglomération dirrectionnelle pour le calcul d'écoulements turbulents.* PhD thesis, Université de Nice-Sophia Antipolis, Nice, France, 1998. (Cited on page 33.)

[Gad-el Hak 1998] Mohamed Gad-el Hak, Andrew Pollard and Jean-Paul Bonnet. Flow control: fundamentals and practices, volume 23. Springer Berlin, 1998. (Cited on page 3.)

[Garnier 2012] E Garnier, PY Pamart, J Dandois and P Sagaut. *Evaluation of the unsteady RANS capabilities for separated flows control.* Computers & Fluids, vol. 61, pages 39–45, 2012. (Cited on pages 9 and 173.)

[Gatski 1993] TB Gatski and CG Speziale. *On explicit algebraic stress models for complex turbulent flows.* Journal of Fluid Mechanics, vol. 254, no. 1, pages 59–78, 1993. (Cited on page 24.)

[Gatski 2007] Thomas B Gatski, Christopher L Rumsey and Rémi Manceau. *Current trends in modelling research for turbulent aerodynamic flows.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 365, no. 1859, pages 2389–2418, 2007. (Cited on page 19.)

[Gibbs 1997] N. Gibbs Mark. *Bayesian Gaussian Processes for Regression and Classification.* PhD thesis, University of Cambridge, 1997. (Cited on pages 93, 94 and 99.)

[Gilarranz 2002] JL Gilarranz, LW Traub and OK Rediniotis. *Characterization of a compact, high power synthetic jet actuator for flow separation control.* AIAA paper, vol. 127, no. 2002, page 12, 2002. (Cited on page 24.)

[Ginsbourger 2009] David Ginsbourger. *Multiples métamodèles pour l'approximation et l'optimisation de fonctions numériques multivariables.* PhD thesis, École Nationale Supérieure des Mines de Saint-Étienne, March 2009. (Cited on pages 93 and 99.)

[Glezer 2002] Ari Glezer and Michael Amitay. *Synthetic jets.* Annual Review of Fluid Mechanics, vol. 34, no. 1, pages 503–529, 2002. (Cited on page 25.)

[Glezer 2004] Ari Glezer. *Fluidic-based virtual aerosurface shaping.* Rapport technique, DTIC Document, 2004. (Cited on pages 3 and 4.)

[Goldberg 1989] David E. Goldberg. Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st édition, 1989. (Cited on pages 5 and 91.)

[Hansen 2006] N. Hansen. *The CMA evolution strategy: a comparing review.* In J.A. Lozano, P. Larranaga, I. Inza and E. Bengoetxea, editeurs, Towards a new evolutionary computation. Advances on estimation of distribution algorithms, pages 75–102. Springer, 2006. (Cited on page 105.)

[Hascoet 2013] Laurent Hascoet and Valérie Pascual. *The Tapenade Automatic Differentiation tool: principles, model, and specification.* ACM Transactions on Mathematical Software (TOMS), vol. 39, no. 3, page 20, 2013. (Cited on page 91.)

[Hirsch 1990] C. Hirsch. Numerical computation of internal and external flows: computational methods for inviscid and viscous flows. Wiley, 1990. (Cited on page 43.)

[Hirsch 2007] Charles Hirsch. Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics, volume 1. Butterworth-Heinemann, 2007. (Cited on page 12.)

[Hoarau 2006] Yannick Hoarau, Marianna Braza, Y Ventikos and D Faghani. *First stages of the transition to turbulence and control in the incompressible detached flow around a NACA0012 wing.* International journal of heat and fluid flow, vol. 27, no. 5, pages 878–886, 2006. (Cited on page 3.)

[Holman 2005] Ryan Holman, Yogen Utturkar, Rajat Mittal, Barton L Smith and Louis Cattafesta. *Formation criterion for synthetic jets.* AIAA journal, vol. 43, no. 10, pages 2110–2116, 2005. (Cited on page 25.)

[Huang 2006] D. Huang, T.T. Allen, W.I. Notz and N. Zeng. *Global optimization of stochastic black-box systems via sequential kriging meta-models.* Journal of Global Optimization, vol. 34, no. 3, pages 441–466, 2006. (Cited on pages 93 and 102.)

[J. Forrester 2006] Alexander I J. Forrester, Andy J Keane and Neil W Bressloff. *Design and analysis of" Noisy" computer experiments.* AIAA journal, vol. 44, no. 10, pages 2331–2339, 2006. (Cited on page 186.)

[Jameson 1991] A. Jameson. *Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings.* AIAA paper, vol. 91, page 1596, 1991. (Cited on page 41.)

[Jameson 2003] Antony Jameson. *Aerodynamic Shape Optimization Using the Adjoint Method.* In VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics, Rhode St Genese, 2003. (Cited on page 91.)

[Jamil 2013] Momin Jamil and Xin-She Yang. *A literature survey of benchmark functions for global optimisation problems.* International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, pages 150–194, 2013. (Cited on page 110.)

[Janusevskis 2013] Janis Janusevskis and Rodolphe Le Riche. *Simultaneous kriging-based estimation and optimization of mean response.* Journal of Global Optimization, vol. 55, no. 2, pages 313–336, 2013. (Cited on page 186.)

[Jasak 1996] Hrvoje Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows.* PhD thesis, Imperial College London (University of London), 1996. (Cited on page 156.)

[Jones 1998] D.R. Jones, M. Schonlau and W.J. Welch. *Efficient global optimization of expensive black-box functions.* Journal of Global optimization, vol. 13, no. 4, pages 455–492, 1998. (Cited on pages 5, 92, 93, 102 and 110.)

[Jones 2001] D.R. Jones. *A taxonomy of global optimization methods based on response surfaces.* Journal of Global Optimization, vol. 21, no. 4, pages 345–383, 2001. (Cited on pages 91, 92, 93 and 101.)

[Kim 2005] Kyu Hong Kim and Chongam Kim. *Accurate, efficient and monotonic numerical methods for multi-dimensional compressible flows: Part II: Multi-dimensional limiting process.* Journal of Computational Physics, vol. 208, no. 2, pages 570–615, 2005. (Cited on page 35.)

[Kloczko 2006] T. Kloczko. *Développement d'une méthode implicite sans matrice pour la simulation 2D-3D des écoulements compressibles et faiblement compressibles en maillages non-structurés.* PhD thesis, Arts et Métiers ParisTech, 2006. (Cited on page 42.)

[Kral 1997] Linda D Kral, John F Donovan, Alan B Cain and Andrew W Cary. *Numerical simulation of synthetic jet actuators.* AIAA paper, vol. 1824, page 1997, 1997. (Cited on pages 26 and 27.)

[Larrouturou 1989] Bernard Larrouturou. *How to preserve the mass fractions positivity when computing compressible multi-component flows.* Rapport de recherche RR-1080, INRIA, 1989. (Cited on page 39.)

[Launder 1974] B.E. Launder and D.B. Spalding. *The numerical computation of turbulent flows.* Computer Methods in Applied Mechanics and Engineering, vol. 3, no. 2, pages 269 – 289, 1974. (Cited on page 21.)

[Laurenceau 2008] Julien Laurenceau. *Surfaces de réponse par krigeage pour l'optimisation de formes aérodynamiques.* PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2008. (Cited on page 92.)

[Le 1997] H. Le, P. Moin and J. Kim. *Direct numerical simulation of turbulent flow over a backward-facing step.* Journal of Fluid Mechanics, vol. 330, no. 1, pages 349–374, 1997. (Cited on page 156.)

[Lesieur 2005] Marcel Lesieur. Large-eddy simulations of turbulence. Cambridge University Press, 2005. (Cited on page 4.)

[MacCormack 1989] Robert W MacCormack and Graham V Candler. *The solution of the Navier-Stokes equations using Gauss-Seidel line relaxation.* Computers & fluids, vol. 17, no. 1, pages 135–150, 1989. (Cited on page 39.)

[MacKay 1992] D.J.C. MacKay. *Bayesian interpolation.* Neural computation, vol. 4, no. 3, pages 415–447, 1992. (Cited on page 98.)

[MacKay 1998] D.J.C. MacKay. *Introduction to Gaussian processes.* NATO ASI Series F Computer and Systems Sciences, vol. 168, pages 133–166, 1998. (Cited on page 93.)

[Mallinson 1999] SG Mallinson, G Hong and JA Reizes. *Some characteristics of synthetic jets.* AIAA paper, vol. 3651, page 1999, 1999. (Cited on page 27.)

[Menter 1993] FR Menter. *Zonal two-equation k−\\ omega turbulencemodelsforaerodynamicflows.AIAAJournal*, 1993.(*Citedonpage* 22.)

[Michalewicz 1996] Zbigniew Michalewicz. Genetic algorithms+ data structures= evolution programs. springer, 1996. (Cited on pages 5, 91 and 104.)

[Mittal 2001] R Mittal, P Rampunggoon and HS Udaykumar. *Interaction of a synthetic jet with a flat plate boundary layer.* AIAA paper, vol. 2773, no. 200, page 1, 2001. (Cited on page 25.)

[Mohammadi 1994] B. Mohammadi and O. Pironneau. Analysis of the k-epsilon turbulence model. RAM: Research in Applied Mathematics. Masson, Paris, 1994. (Cited on page 16.)

[Mohammadi 2001] Bijan Mohammadi, Olivier Pironneau, B Mohammadi and Oliver Pironneau. Applied shape optimization for fluids, volume 28. Oxford University Press Oxford, 2001. (Cited on page 91.)

[Pes 2002] Matteo Pes, Bojan Lukovic, Paul Orkwis and M Turner. *Modeling of two dimensional synthetic jet unsteadiness using neural network-based deterministic source terms.* AIAA Paper, vol. 2860, page 2002, 2002. (Cited on page 27.)

[Picheny 2011] Victor Picheny, Tobias Wagner and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. 2011. (Cited on pages 93, 103, 107 and 114.)

[Picheny 2012] Victor Picheny, David Ginsbourger*et al. Approximating computer experiments using partially converged simulations and a Gaussian process emulator.* 2012. (Cited on page 186.)

[Picheny 2013] V. Picheny and D. Ginsbourger. *A Nonstationary Space-Time Gaussian Process Model for Partially Converged Simulations.* SIAM/ASA Journal on Uncertainty Quantification, vol. 1, no. 1, pages 57–78, 2013. (Cited on page 186.)

[Pironneau 1974] Olivier Pironneau. *On optimum design in fluid mechanics.* Journal of Fluid Mechanics, vol. 64, no. 01, pages 97–110, 1974. (Cited on page 91.)

[Prandtl 1904] Ludwig Prandtl. *Uber Flussigkeits bewegung bei sehr kleiner Reibung.* Verhaldlg III Int. Math. Kong, pages 484–491, 1904. (Cited on page 3.)

[Queutey 2007] P. Queutey and M. Visonneau. *An interface capturing method for free-surface hydrodynamic flows.* Computers & fluids, vol. 36, no. 9, pages 1481–1510, 2007. (Cited on page 156.)

[Rasmussen 2006] Carl Edward Rasmussen. *Gaussian processes for machine learning.* 2006. (Cited on pages 95 and 98.)

[Rhie 1982] C. Rhie and W. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. American Institute of Aeronautics and Astronautics, 2014/09/03 1982. (Cited on page 156.)

[Ritchie 2000] BD Ritchie, DR Mujumdar and JM Seitzman. *Mixing in coaxial jets using synthetic jet actuators.* AIAA paper, vol. 404, page 2000, 2000. (Cited on page 3.)

[Rizzetta 1999] Donald P Rizzetta, Miguel R Visbal and Michael J Stanek. *Numerical investigation of synthetic-jet flowfields.* AIAA journal, vol. 37, no. 8, pages 919–927, 1999. (Cited on page 25.)

[Roe 1981] Philip L Roe. *Approximate Riemann solvers, parameter vectors, and difference schemes.* Journal of computational physics, vol. 43, no. 2, pages 357–372, 1981. (Cited on page 38.)

[Rumsey 2006] Christopher L Rumsey, TB Gatski, WL Sellers III, VN Vasta and SA Viken. *Summary of the 2004 computational fluid dynamics validation workshop on synthetic jets.* AIAA Journal, vol. 44, no. 2, pages 194–207, 2006. (Cited on pages 10 and 27.)

[Rumsey 2008] C.L. Rumsey and J.L. Thomas. *Application of FUN3D and CFL3D to the Third Workshop on CFD Uncertainty Analysis.* In Proceedings of the 3rd Workshop on CFD Uncertainty Analysis, 2008. (Cited on page 156.)

[Rusanov 1961] V.V. Rusanov. *Calculation of interaction of non-steady shock waves with obstacles.* J. Comput. Math. Phys. USSR, vol. 1, pages 267–279, 1961. (Cited on page 42.)

[Sacks 1989] J. Sacks, W.J. Welch, T.J. Mitchell and H.P. Wynn. *Design and analysis of computer experiments.* Statistical science, vol. 4, no. 4, pages 409–423, 1989. (Cited on page 173.)

[Sagaut 2002] Pierre Sagaut. Large eddy simulation for incompressible flows. Springer, 2002. (Cited on page 4.)

[Seifert 1996] A Seifert, A Darabi and I Wyganski. *Delay of airfoil stall by periodic excitation.* Journal of Aircraft, vol. 33, no. 4, pages 691–698, 1996. (Cited on pages 3, 24 and 153.)

[Seifert 1999] Avi Seifert and LaTunia G Pack. *Oscillatory Excitation of Unsteady Compressible Flows over Airfoils at Flight Reynolds Numbers*. 1999. (Cited on page 153.)

[Smith 1997] Barton L Smith and Ari Glezer. *Vectoring and small-scale motions effected in free shear flows using synthetic jet actuators*. AIAA paper, vol. 213, page 1997, 1997. (Cited on pages 3 and 24.)

[Spalart 1986] Philippe R Spalart. *Numerical study of sink-flow boundary layers*. Journal of Fluid Mechanics, vol. 172, pages 307–328, 1986. (Cited on page 4.)

[Spalart 1992] P.R. Spalart and S.R. Allmaras. *A one-equation turbulence model for aerodynamic flows*. In AIAA, Aerospace Sciences Meeting and Exhibit, 30 th, Reno, NV, page 1992, 1992. (Cited on page 20.)

[Spalart 2009] Philippe R Spalart. *Detached-eddy simulation*. Annual Review of Fluid Mechanics, vol. 41, pages 181–202, 2009. (Cited on page 4.)

[Spalart 2012] P. Spalart. *Reflections on RANS Modeling*, June-August 2012. (Cited on page 16.)

[Steger 1981] Joseph L Steger and R.F Warming. *Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods*. Journal of Computational Physics, vol. 40, no. 2, pages 263 – 293, 1981. (Cited on pages 36 and 39.)

[Stein 1999] Michael L Stein. Interpolation of spatial data: some theory for kriging. Springer, 1999. (Cited on pages 97 and 98.)

[Toro 1994] EF Toro, M. Spruce and W. Speares. *Restoration of the contact surface in the HLL-Riemann solver*. Shock waves, vol. 4, no. 1, pages 25–34, 1994. (Cited on page 36.)

[Toro 2009] Eleuterio F Toro. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer, 2009. (Cited on page 15.)

[Utturkar 2002] Yl Utturkar, R Mittal, P Rampunggoon and L Cattafesta. *Sensitivity of synthetic jets to the design of the jet cavity*. AIAA paper, vol. 124, page 2002, 2002. (Cited on page 25.)

[Vadillo 2002] Jose L Vadillo, Ramesh K Agarwal, Andrew W Cary and William W Bower. *Numerical study of virtual aerodynamic shape modification of an airfoil using a synthetic jet actuator*. PhD thesis, Washington University, 2002. Department of Mechanical Engineering., 2002. (Cited on page 27.)

[Van Leer 1979] Bram Van Leer. *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*. Journal of computational Physics, vol. 32, no. 1, pages 101–136, 1979. (Cited on page 35.)

[Vazquez 2010] Emmanuel Vazquez and Julien Bect. *Convergence properties of the expected improvement algorithm with fixed mean and covariance functions.* Journal of Statistical Planning and Inference, vol. 140, no. 11, pages 3088 – 3095, 2010. (Cited on page 102.)

[Viozat 2001] C. Viozat, C. Held, K. Mer and A. Dervieux. *On vertex-centered unstructured finite-volume methods for stretched anisotropic triangulations.* Computer methods in applied mechanics and engineering, vol. 190, no. 35, pages 4733–4766, 2001. (Cited on page 34.)

[Wieghardt 1951] Karl Wieghardt and W Tillman. *On the turbulent friction layer for rising pressure.* 1951. (Cited on page 50.)

[Williams 1998] C.K.I. Williams. *Prediction with Gaussian processes: From linear regression to linear prediction and beyond.* NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES, vol. 89, pages 599–621, 1998. (Cited on page 93.)

[Wright 1999] SJ Wright and J Nocedal. Numerical optimization, volume 2. Springer New York, 1999. (Cited on page 91.)

[Xia 2005] Hao Xia and Ning Qin. *Dynamic grid and unsteady boundary conditions for synthetic jet flow.* AIAA, vol. 106, page 2005, 2005. (Cited on page 25.)

[Yamaleev 2005] Nail K Yamaleev, Mark H Carpenter and Frederick Ferguson. *Reduced-order model for efficient simulation of synthetic jet actuators.* AIAA journal, vol. 43, no. 2, pages 357–369, 2005. (Cited on page 10.)

[Yoder 1999] D.A. Yoder and N.J. Georgiadis. Implementation and validation of the chien k-[epsilon] turbulence model in the wind navier-stokes code. National Aeronautics and Space Administration, Glenn Research Center, 1999. (Cited on page 156.)

[Yoon 2008] Sung-Hwan Yoon, Chongam Kim and Kyu-Hong Kim. *Multi-dimensional limiting process for three-dimensional flow physics analyses.* Journal of Computational Physics, vol. 227, no. 12, pages 6001–6043, 2008. (Cited on page 35.)

Optimization of active control devices for separated turbulent flows

Abstract: Active flow control strategies, such as oscillatory blowing / suction, have proved their efficiency to modify flow characteristics for various purposes (e.g. skin friction reduction, separation delay, etc.) in case of rather simple configurations. To extend this approach to industrial cases, the simulation of a large number of devices at real scale and the optimization of parameters are required. The objective of this thesis is to set up an optimization procedure to solve this category of problems. In this perspective, the organization of the thesis is split into three main parts.

First, the development and validation of an unsteady compressible turbulent flow solver using the Reynolds-Averaged Navier-Stokes (RANS) using a Mixed finite-Element/finite-Volume (MEV) framework is described. A particular attention is drawn on synthetic jet numerical model implementation by comparing different models in the context of a simulation over a flat plate.

The second axis of the thesis describes and validates the implementation of a Gaussian Process surrogate model based global optimization method including an approach to account for some numerical errors during the optimization. This EGO (Efficient Global Optimization) method, is validated on noisy 1D and 2D analytical test cases.

Finally, the optimization of two industrial relevant test cases is considered. The first case is a compressible turbulent flow over a NACA0015. The time-averaged lift is regarded as the control criterion to be maximized. The second one is an incompressible turbulent flow over a Backward Facing Step for which the time-averaged recirculation length is minimized. Both test cases include a synthetic jet actuator as a control device to be optimized. The EGO method is shown to be very efficient given a well determined observation error variance. It is also demonstrated that numerical parameters (grid size, time step, etc.) may have a significant influence on the results, while the choice of the turbulence closure is critical.

The simulations are carried out with the in-house compressible flow solver NUM3SIS developed at INRIA and with the incompressible flow solver ISIS-CFD developed by the CFD group of the Fluid Mechanics Laboratory at École Centrale de Nantes (ECN).

Keywords: Backward Facing Step, Control device, Efficient Global Optimization, Flow control, Gaussian Process, Kriging model, Mixed finite-Element/finite-Volume (MEV), NACA0015, Synthetic Jet, Turbulence

Optimisation de dispositifs de contrôle actif pour des écoulements turbulents décollés

Résumé: Les stratégies de contrôle d'écoulement, telles que le soufflage / aspiration, ont prouvé leur efficacité à modifier les caractéristiques d'écoulement à des fins diverses en cas de configurations usuellement simples. Pour étendre cette approche sur des cas industriels, la simulation de dispositifs à échelle réelle et l'optimisation des paramètres de contrôle s'avèrent nécessaires. L'objectif de cette thèse est de mettre en place une procédure d'optimisation pour résoudre cette catégorie de problèmes. Dans cette perspective, l'organisation de la thèse est divisé en trois parties.

Tout d'abord, le développement et la validation d'un solveur d'écoulement turbulent compressible instationnaire, résolvant les équations de Navier-Stokes moyennées (RANS) dans le cadre d'une discrétisation mixte de type éléments finis / volumes finis (MEV) sont présentés. Une attention particulière est portée sur la mise en oeuvre de modèles numériques de jet synthétique à l'aide de simulations sur une plaque plane.

Le deuxième axe de la thèse décrit et valide la mise en oeuvre d'une méthode d'optimisation globale basée sur un modèle réduit du type processus gaussien (GP), incluant une approche de filtrage d'erreurs numériques liées aux observations. Cette méthode EGO (Efficient Global Optimization), est validée sur des cas analytiques bruités 1D et 2D.

Pour finir, l'optimisation de paramètres de contrôle de jet synthétique sur deux cas test pertinents pour les industriels est considérée. Le premier cas est un écoulement turbulent compressible autour d'un profil d'aile NACA0015, dont la portance moyenne est l'objectif à maximiser. Le second cas est un écoulement turbulent incompressible sur une marche descendante, pour lequel on souhaite minimiser la longueur moyenne de recirculation.

Les simulations sont effectuées avec le solveur d'écoulement compressible NUM3SIS développé au sein d'INRIA et avec le solveur incompressible ISIS-CFD développé par le groupe CFD du Laboratoire de Mécanique des Fluides de l'École Centrale de Nantes.

Mots-clés: Marche descendante, Dispositif de contrôle, Efficient Global Optimization, Contrôle d'écoulement, Processus Gaussien, Modèle de krigeage, Éléments finis/Volumes finis mixtes (MEV), NACA0015, Jet synthétique, Turbulence

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE SFA

SCIENCES FONDAMENTALES ET APPLIQUÉES

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis

Discipline : Mathématiques Appliquées

Présentée et soutenue par

Jérémie Labroquère

Optimisation de dispositifs de contrôle actif pour des écoulements turbulents décollés

Directeurs de Thèse : Jean-Antoine Désidéri & Régis Duvigneau, équipe-projet OPALE

préparée à INRIA Sophia Antipolis, équipe-projet OPALE

soutenue le 20 novembre 2014

Jury :

Rapporteurs :	Christian Tenaud	- DR -	CNRS UPR 3251 - LIMSI
	Michel Visonneau	- DR -	CNRS UMR 6598 - LMF - ECN
Directeurs :	Jean-Antoine Désidéri	- DR -	INRIA - OPALE
	Régis Duvigneau	- CR -	INRIA - OPALE
Examinateurs :	Marianna Braza	- DR -	CNRS UMR 5502 - IMFT
	Rodolphe Le Riche	- CR -	CNRS UMR 6158 - LIMOS - EMSE

Introduction translated in French:

Durant les dernières décennies, l'industrie de l'aviation a littéralement explosé et est devenu un domaine très concurrentiel. Depuis, de nouveaux problèmes menaçant l'environnement tel que l'activité humaine et le réchauffement climatique ont été soulevés. En Europe, cette question a été définie comme l'une des priorités de l'industrie de l'aviation pour les 40 prochaines années. En effet, la vision de l'Europe de l'aviation vise pour 2050 une réduction de 75% d'émissions de CO_2, de 90% en NO_x et de 65% l'émission de bruit perçu pour les aéronefs en vol par rapport à la capacité d'un nouvel avion type de 2000. Pour atteindre de telles réductions, accroître l'efficacité des technologies actuelles ne sera certainement pas suffisant.

Les nouvelles technologies telles que le contrôle d'écoulement actif a été mis en avant par les industries, tels que Airbus, comme une technologie clé afin de répondre aux exigences d'émissions de gaz à effet de serre d'ici à 2050. Le contrôle d'écoulement consiste à modifier l'état d'un écoulement naturel pour atteindre un nouvel état ayant des propriétés intéressantes. En d'autres termes, les actionneurs sont introduites dans un écoulement pour obtenir des caractéristiques souhaitables. Ce n'est pas une idée récente, au début du siècle dernier, Ludwig Prandtl a utilisé le contrôle d'écoulement pour expliquer sa théorie de la couche limite en supprimant complètement la séparation de l'écoulement derrière d'un cylindre circulaire grâce à l'aide aspiration constante.

Le contrôle d'écoulement peut être passive ou actif. Le contrôle d'écoulement passif est obtenu en utilisant des actionneurs qui ne nécessitent pas d'alimentation externe pour fonctionner. Ces actionneurs sont généralement des installations fixes mécaniques agissant sur la couche limite, comme des riblets disposés sur la surface, des générateurs de vortex, etc. Ils sont faciles à mettre en oeuvre, mais leur efficacité est limitée et ils souffrent d'un manque de flexibilité.

Le contrôle d'écoulement actifs, d'autre part, utilise une source d'énergie supplémentaire. L'injection de momentum en continu, par soufflage ou aspiration de fluide fait partie de cette catégorie. Malheureusement, le soufflage de fluide en continu a un rendement faible et nécessite des systèmes d'alimentation d'air, ce qui l'éloigne en pratique de l'utilisation industrielle. Alternativement, les actionneurs actifs exploitent les phénomènes d'instabilités naturelles des écoulements et est un domaine qui a reçu un grand intérêt dans la recherche au cours de ces dernières décennies, et ce, depuis que leur capacité à améliorer les performances aérodynamiques ait été démontré. Un contrôle efficace peut être réalisé en utilisant des actionneurs périodiques, qui peuvent interagir avec les structures cohérentes à grande échelle. Ce type excitations d'écoulement périodiques peuvent être générés en utilisant des jets synthétiques, actionneurs plasma, actionneurs de décharge à barrière diélectrique, etc. Plusieurs expériences ont été menées avec succès, de façon expérimentale ou numérique, afin d'appliquer cette technologie dans divers cas, tels que le retard de séparation, l'amélioration de mélange, la vectorisation d'écoulement, etc.

Une difficulté majeure dans le contrôle d'écoulement actif est le choix des paramètres de

contrôle tels que l'amplitude, la fréquence, etc. De plus, mener une étude complète avec des études expérimentales uniquement sur une telle échelle est inconcevable. En effet, avec un avion réel, des centaines d'actionneurs seraient nécessaires et faire le choix des paramètres à la main n'est pas possibles.

Dernièrement, les capacités numériques et de calcul des algorithmes de mécanique des fluides ont atteint un niveau de maturité qui permet de mener des expériences numériques, remplacant les campagnes régulières de soufflerie et introduisant la validation des modèles à un stade plus précoce dans le processus de conception. Comme le but est d'exploiter les phénomènes d'instabilité naturelle pour manipuler les caractéristiques d'écoulement, le contrôle d'écoulement actif est très sensible aux paramètres de l'actionneur. Trouver les meilleurs paramètres efficaces actionneurs est donc une exigence. Bien que de nombreuses études ont porté sur des études paramétriques, les algorithmes d'optimisation avancés gagnent un intérêt car ils peuvent aider à déterminer efficacement ces paramètres. En couplant simulation d'écoulements avec actionneurs et des algorithmes d'optimisation, on pouvait espérer pouvoir déterminer des paramètres de contrôle efficaces pour un coût raisonnable.

Malheureusement, la simulation est encore limitée. La première difficulté est liée à la modélisation de la turbulence. Trois niveaux de modélisation de la turbulence sont utilisés en pratique. Le premier est la DNS, pour Direct Numerical Simulation. Cette modélisation ne fait aucune hypothèse sur la turbulence physique et résout les plus petits tourbillons existant dans l'écoulement. Ce niveau de modélisation est mathématiquement insoluble pour les écoulements industriels considérés en raison des exigences en terme de maillage. Le deuxième niveau de la modélisation est la LES / DES, pour Large Eddy Simulation ou Detached Eddy Simulation. Ils modélisent les petites échelles de tourbillons et ne font aucune hypothèse sur les grandes échelles. Ces deux modèles sont plus abordables en termes de coût de CPU, mais leurs besoins sont encore de l'ordre de quelques semaines pour une seule simulation. En outre, leur applicabilité dans l'industrie est encore problématique en raison des exigences en termes de maillage et coût de calcul. Le dernier niveau de modélisation est le RANS, pour Reynolds Averaged Navier-Stokes Equations. Cette technique de modélisation consiste à appliquer un opérateur de filtrage sur les équations de Navier-Stokes, afin pour résoudre l'écoulement moyen seulement. Ce modèle est le plus abordable et est couramment utilisé dans l'industrie. Le principal inconvénient est que la plupart du temps, la physique n'est pas correctement reproduites si les simulations sont exécutées trop loin des cas de calibration des modèles. Malgré ces inconvénients, de nombreuses simulations d'écoulements contrôlés instationnaires en utilisant des modèles RANS peuvent être trouvés dans la littérature.

En ce qui concerne l'optimisation, les principales difficultés dans le cadre du contrôle d'écoulement turbulent est le temps CPU et les possible optima locaux. Les méthodes d'optimisation à base de gradient telles que l'algorithme du gradient, peut réduire le nombre d'évaluations nécessaires pour optimisation complète. Cela conduit à un temps de CPU faible, mais ces méthodes sont connues pour être prises au piège dans un optimum local. En outre, le calcul des gradients de la fonction objectif par rapport aux paramètres

de contrôle, exclut l'utilisation des différences finie, en raison du coût de l'évaluation du solveur. Les approches Adjointes sont généralement utilisées pour calculer efficacement ces quantités avec des écoulements stationnaires, mais ils deviennent lourd et difficile à appliquer en cas d'écoulements instationnaires. Les méthodes méta-heuristiques (Evolution Stratégie, Algorithme Génétique, essaim de particules, etc.) peuvent atteindre un optimum global mais sont trop coûteux car ils nécessitent des centaines de milliers d'évaluations de la fonction.

Par conséquent, l'objectif de cette thèse est d'étudier la faisabilité d'une technique d'optimisation globale alternative basée sur des simulations instationnaires pour déterminer les paramètres d'actuation dans le cadre des écoulements turbulents. Plus précisément, nous allons nous concentrer sur l'approche d'optimisation globale efficace (EGO), qui repose sur une modélisation statistique adaptative de la fonctionnelle, permettant de tenir compte des erreurs d'observation. En outre, une attention particulière sera accordée à l'impact la modélisation physique dans le processus d'optimisation. La capacité des modèles RANS pour déterminer les paramètres de contrôle efficaces est étudiée dans ce contexte.

Ce manuscrit est structuré comme il suit. Le chapitre 1 présente les méthodes de calcul de mécanique des fluides avec une perspective d'optimisation. Le chapitre 2 présente la description de la procédure d'optimisation dans la perspective d'utiliser des simulations très coûteuses pour évaluer la fonction coût. Le chapitre 3 présente l'application de ces méthodes sur les problèmes industriellement pertinents de contrôle d'écoulement actif (proposés dans le cadre du projet MARS).

Conclusion translated in French :

Dans cette thèse, une combinaison de méthodes numériques a été présenté afin de s'attaquer à l'optimisation de dispositifs de contrôle actif pour les écoulements turbulents séparés. Ce problème est difficile en essence car il fait face à diverses disciplines, telles que la simulation numérique, la modélisation de la turbulence, le calcul haute performance, l'optimisation, etc. Les limites et les difficultés pour appliquer des méthodes d'optimisation classiques, comme les méthodes heuristiques ou de gradient. Elles sont principalement dues à la multi-modalité, l'instationnarité, ce qui conduit à l'évaluation du gradient de la fonction coût ardue en utilisant les méthodes adjointes et des restrictions sur le temps CPU.

Coupler l'optimisation avec des simulations d'écoulement turbulent instaionnaire est un grand défi, car elle introduit des exigences rigides et des contraintes en termes de techniques d'optimisation et de modélisation physique. Le choix a été porté sur les modèles RANS, en raison du coût de calcul, et sur l'optimisation à base de méta-modèle, pour son efficacité et sa capacité de recherche globale.

Dans cette thèse, les méthodes de calcul pour la simulation d'écoulements turbulent compressible instationnaire suivis par des méthodes d'optimisation basées sur les méta-modèles ont été décrits en premier lieu. La variance de l'observation a été introduite dans la génération de méta-modèle pour prendre en compte les incertitudes d'observation de la fonction de coût. Toutes les méthodes mises en oeuvre ont été soigneusement testés sur un certain nombre de cas de tests pertinents pour les problèmes considérés, avant d'envisager deux cas de test d'optimisation : le NACA0015 et la marche descendante.

La stratégie d'EGO appliqué à la maximisation de la portance d'un profil d'aile NACA0015 à 18 degrés et à la réduction de la longueur de recirculation après une marche descendante s'est révélé être vraiment efficace pour trouver les paramètres optuma du jet synthétique. Dans les deux cas, les exercices d'optimisation ont été réalisées avec différentes fermetures de turbulence de type RANS. Pour le NACA0015, après environ 15 évaluations du solveur, une augmentation de la portance d'environ 20% a été observé pour les fermetures de turbulence Spalart-Allmaras et k-Ω SST. La réduction de la longueur de recirculation après une marche descendante en utilisant l'optimiseur, était en moyenne de 73% de la longueur de recirculation initiale, après 30 évaluations d'optimisation pour les fermetures de turbulence Spalart-Allmaras, k-ε Launder-Sharma, k-ω SST Menter et k-ω-ne-easm.

Ces exercices d'optimisation ont montré la nécessité de tenir compte des erreurs numériques lors de l'optimisation à base de méta-modèle. En effet, l'utilisation de modèles d'interpolation donne des représentations oscillantes et irréalistes, ce qui conduit à l'échec du processus d'optimisation. L'introduction de la notion d'observation bruité a permis de surmonter cette difficulté. La provenance de l'erreur numérique a été identifié comme venant de diverses étapes du processus de simulation, comme le maillage, l'ordre de réduction des résidus, etc.

En dehors de l'incertitude introduite par l'erreur numérique, il a été trouvé que la modélisation de la turbulence a le plus d'impact, donnant sensiblement différents paramètres optimaux. En outre, les quantités d'intérêt présentent des variations de 25% lorsque les différentes fermetures ont été testés. Malgré ces différences, des similitudes entre les écoulements avec les paramètres de contrôle optimal ont été signalés, à la fois pour le contrôle du NACA0015 et la marche descendante, ce qui indique que les modèles RANS peuvent encore être utilisés pour déterminer des encadrements de paramètres de contrôle d'écoulement efficaces.

Beaucoup de questions sont encore ouvertes afin d'étendre l'optimisation basée sur les écoulements turbulents instationnaires à des cas industriels plus complexes.

Le premier concerne le point de vue et la gestion des erreurs au cours de l'optimisation. En effet, il a été montré dans ce contexte que l'estimation de l'erreur a été nécessaire pour assurer optimisation réussie. L'estimation de l'erreur qui a été faite expérimentalement en augmentant le niveau de variance jusqu'à obtention d'un méta-modèle satisfaisant. Le problème d'estimer correctement l'erreur s'avère être l'aspect certainement le plus complexe . Pour gérer plus de deux paramètres, avoir d'une manière automatique pour régler la variance d'observation, est d'autant plus nécessaire qu'il n'y a pas possibilité d'observer le méta-modèle.

Une autre perspective concernerait la prise en compte des incertitudes physiques lors de l'optimisation. En effet, l'utilisation pratique des actionneurs sera soumise à plusieurs incertitudes, découlant des conditions d'écoulement (nombre de Reynolds, nombre de Mach, incidence, etc.). Dans ce contexte, une optimisation robuste de paramètres de l'actionneur doit être atteinte, qui consiste à minimiser la moyenne la fonction de coût.

La deuxième question ouverte est de surmonter les limitations introduites par la modélisation RANS. L'extension de la méthode à des modèles hybrides RANS / LES est simple du point de vue informatique, mais implique une explosion du temps CPU. Un bon candidat dans la perspective de résoudre ce problème ambitieux, est d'utiliser des solutions partiellement convergées de façon à réduire légèrement les exigences. Bien sûr, cela ne peut se faire que si le premier point (de gestion de la variance) est résolu. Une dernière perspective serait d'avoir un algorithme adaptatif, qui règle automatiquement la longueur de simulation requise, et le mélanges de simulations courtes avec une grande variance et de longues simulations avec une variance faible.

Le travail effectué au cours de cette thèse a mis en place les fondations pour envisager un large panel de nouvelles applications. D'un point de vue de la simulation d'écoulements, l'extension de l'optimisation pour des cas 3D est bien sûr l'un des premiers exercices qui aurait vraiment de sens pour l'industrie. Ensuite, l'extension de l'optimisation avec un grand nombre de paramètres à optimiser pourrait être un pas en avant pour résoudre des problèmes réalistes. Cela pourrait inclure la position de l'actionneur, son orientation, etc. Enfin, différents actionneurs que les jets synthétiques pourraient être envisagés. En effet, un grand nombre de rapports expérimentaux et numériques sont disponibles à ce

jour utilisant des méthodes alternatives, telles que les actionneurs plasma DBD, ou des jets pulsés.

Development translated in French:

Cette partie regroupe une traduction partielle des sous conclusions tirées dans cette thèse. Pour de plus ample détails, merci de vous reporter à la version écrite en anglais.

La validation de la mise en œuvre de jet synthétique :

L'objectif de cette étude était de simuler un jet synthétique dans un écoulement de type couche limite turbulente, et de comparer les prédictions de l'écoulement généré par certains modèles d'actionneurs ainsi qu'évaluer les fermetures de la turbulence dans le contexte présent. Il a été constaté que le modèle de l'actionneur comprenant la description géométrique de la conduite est un bon compromis comparé à la complexité introduite avec un maillage complet de la cavité et dont la simplicité de mise en place est de mise en utilisant d'une simple condition aux limites. Les paramètres numériques (critère de convergence, la taille du maillage, pas de temps, type de condition aux limites) ont été sélectionnés à des valeurs raisonnables pour le problème considéré. En ce qui concerne l'influence de la fermeture de la turbulence, un écart modéré entre le modèle de Spalart-Allmaras et les modèles k-ω SST a été rapporté, le dernier modèle générant un écoulement plus intense lors de la phase de soufflage.

Optimisation EGO :

La validation des techniques de l'EGO montre l'importance de l'introduction d'un contrôle de filtrage du bruit lors de l'optimisation des fonctionnelles bruitées. En effet, si le bruit n'est pas considérés, l'optimisation ne converge pas et le méta-modèle commence à osciller au cours des itérations. Dans cette partie, nous avons montré que la mise en oeuvre consistant à modifier la fonction de covariance dans les Processus Gaussiens est extrêmement efficace pour tous les cas tests. En outre, on peut noter que 30 fonctions évaluations sont généralement suffisant pour trouver au moins un minimum pour une optimisation de deux paramètres.

Cas test NACA :

Dans cette étude, une optimisation globale de l'amplitude et de la fréquence d'un jet synthétique a été réalisée. Il a été démontré que la prise en compte du bruit d'observations de la fonction coût est obligatoire afin d'obtenir un méta-modèle correct. Après avoir choisi une erreur de simulation, les fonctions de mérite EI et PI ont été appliquées sur le problème d'optimisation avec succès. Compte tenu du nombre limité d'évaluations, en utilisant la fermeture de la turbulence k-ω SST, la fonction de mérite EI a montré de meilleurs résultats que PI car elle a pu localiser deux paramètres optimums alors que PI n'en a trouvé qu'un seul sans explorer d'autre parties intéressante du domaine de recherche.

Une augmentation de la portance d'environ 20% pourrait être réalisé à la fois pour les fermetures Spalart-Allmaras et k-Ω SST. L'ordre de grandeur obtenu pour amélioration des coefficients de portance au moyen de jets synthétiques sont expérimentale-

ment et numériquement observé pour les configurations post-décrochage, bien que dans les références citées, les jets synthétiques pour le cas NACA0015 sont placés en bord d'attaque et non sur la surface supérieure comme lors de nos expérimentations. Dans ces documents, la fréquence utilisée pour contrôler la portance permet d'obtenir des tourbillons bien détachés qui sont convectés en aval de l'aile. Ce phénomène est très similaire au premier point optimal trouvé en utilisant EI et la fermeture k-ω SST. Ces fréquences ont tendance à exciter la fréquence de le lâcher naturelle (non observé ici à 18 degrés, à l'aide de simulations URANS). La fréquence de le lâcher naturelle pour une configuration NACA0015 à $\alpha = 22\ deg$ à nombre de Reynolds comparable, est reporté d'être $f_{shed}L_{ref}/U_{inf} \approx 0.53$, avec une fréquence optimale d'acuation $f_{jet}L_{TE}/U_{inf} \approx 1$, avec L_{TE} la distance entre le bord de fuite et l'actuateur. Même si le cas considéré ici est pas exactement le même, le premier optimum trouvé par l'EI ($f_{jet}L_{ref}/U_{inf} = 1.14$) donne $f_{jet}L_{TE}/U_{inf} = 1.003$, se révélant être une fréquence d'actuation effective proche de l'unité comme donné par Seifert. Duvigneau et al. rapporte fréquence optimale de $f_{jet}L_{ref}/U_{inf} = 0.85$ et une amplitude de $U_{jet}/U_{inf} = 1.72$ pour l'optimisation d'un synthétique jet sur une aile NACA0015 à 18 degrés à l'aide de le fermeture k-ω SST. Ces paramètres optimums n'on pas pu être observés sur les présents méta-modèles. Cet écart entre le travail présenté et les valeurs optima signalés dans l'article de Duvigneau et al. peut provenir du fait que que l'angle de jet a également été optimisé pour atteindre atteint une valeur de 30 degrés. Seifert, a également rapporté qu'une faible amplitude n'est efficaces que si la fréquence d'actuation de jet $f_{jet}L_{TE}/U_{inf} > 2$. Ce résultat a également pu être remarqué dans nos expériences d'optimisation, en utilisant la fermeture Spalart-Allmaras et la k-ω SST (EI 2), où les paramètres de contrôle optimums trouvés sont $f_{jet}L_{ref}/U_{inf} \approx 2$ et $U_{jet}/U_{inf} \approx 0.5$.

Le mécanisme pour augmenter la portance à faible amplitude et haute fréquence, est le même à la fois pour la fermeture Spalart-Allmaras et k-ω SST en introduisant une petite perturbation dans l'écoulement. D'un autre côté, pour les paramètres de contrôle de moyens, la fermeture k-ω SST fournit un optimum donnant lieu à une génération de grands tourbillons convectés en aval, conduisant à un mécanisme radicalement différent.

Les méta-modèles diffèrent de façon significative pour les deux fermetures de turbulence dans les moyennes fréquences et l'amplitudes, ce qui conduit à un optimum absent dans le méta-modèle en utilisant la fermeture de Spalart-Allmaras. Avec ces résultats, on observe que l'impact de la fermeture de la turbulence peut avoir une grande influence sur le processus d'optimisation. Pour cette raison, une étude particulière de l'impact de la fermeture de la turbulence est réalisée sur la marche descendante et présenté dans le chapitre suivant.

<u>Cas test BFS</u>:

Pour le test-cas étudié ici, les caractéristiques de l'écoulement de base dépendent de manière significative de la fermeture de la turbulence. Ceci est usuel lorsque l'on considère des écoulements séparés. La quantité d'intérêt (longueur de séparation), qui est également le critère d'optimisation pour le contrôle actif, présente des variations de 25%

suivant les différentes fermetures testées. Ceci est évidemment un contexte difficile dans le cadre de l'optimisation. Cependant, il est souligné que l'optimisation peut être réalisée avec succès, en dépit de ces différences, si les différentes fermetures donnent lieu à des d'écoulement similaires, lorsque les les paramètres de contrôle sont modifiés. Si la même erreur de modélisation est produite durant toute la procédure d'optimisation, les différentes fermetures peuvent mener aux mêmes paramètres de optimale, même si les écoulements diffèrent. Cela a déjà été observé dans des exercices d'optimisation de forme liés à la chute de pression par exemple : la forme optimale peut être trouvé même si le champ de pression n'est pas prédire avec précision.

Malheureusement, les résultats présentés ci-dessus montrent que les écoulements séparé avec actuation n'est pas un cadre très sympathique. Les écarts observés pour l'écoulement de base non contrôlé donne lieu à différentes directions de recherche et, pour finir à des paramètres optimaux significativement différents. On peut en conclure que les modèles RANS ne peuvent pas être utilisées aveuglément dans une procédure d'optimisation pour déterminer les paramètres d'actuateurs. De toute évidence, les "vrais" ou paramètres physiques optimaux d'actuation sont inconnus. Pour l'écoulement de base, la fermeture k-ω SST fournit la meilleure prédiction de longueur de recirculation, comparé aux expériences. Néanmoins, il est encore difficile de savoir si les meilleurs paramètres prévus par cette fermeture sont proches du «vraie» optimum. L'utilisation de la LES pourrait fournir des résultats de référence, mais l'optimisation basée sur la LES est toujours inabordables, pour des raisons de temps de calcul.

Cependant, des similitudes entre les écoulement avec une actuation optimale ont été rapportés, à l'exception de la fermeture Spalart-Allmaras. Cela semble indiquer que les différents exercices d'optimisation donnent lieu à des caractéristiques d'écoulement similaires, quelle que soit la fermeture. Si cela est confirmé, les modèles RANS peuvent encore être utilisés dans une première étape de conception, afin de déterminer les caractéristiques d'actuateurs d'écoulement efficaces.