Formulation éléments finis variationnelle adaptative et calcul massivement parallèle pour l'aérothermique industrielle

Alban Bazile

To cite this version:

Alban Bazile. Formulation éléments finis variationnelle adaptative et calcul massivement parallèle pour l'aérothermique industrielle. Mécanique des fluides [physics.class-ph]. Université Paris sciences et lettres, 2019. Français. NNT : 2019PSLEM008 . tel-02420374

HAL Id: tel-02420374
https://pastel.hal.science/tel-02420374

Submitted on 19 Dec 2019

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Variational Adaptive Finite Element Formulation and Massively Parallel Computing for Aerothermal Industry Applications

Formulation éléments finis variationnelle adaptative et calcul massivement parallèle pour l'aérothermique industrielle

Soutenue par
Alban BAZILE
Le 25 Avril 2019

École doctorale n°364
Sciences Fondamentales et Appliquées

Spécialité
Mécanique Numérique et Matériaux

Composition du jury :

Pr. Stéphane DESCOMBES Université Côte d'Azur	*Président*
Dr. Luisa ROCHA DA SILVA École Centrale de Nantes	*Rapporteuse*
Pr. Alvaro L.G.A. COUTINHO Federal University of Rio de Janeiro	*Rapporteur*
Dr. Anca BELME Sorbonne Université	*Examinatrice*
Juan-Carlos LARROYA-HUGUET Safran Aircraft Engines	*Examinateur*
Dr. Youssef MESRI MINES-Paristech	*Directeur de Thèse*
Pr. Elie HACHEM MINES-Paristech	*Directeur de Thèse*

MINES ParisTech | PSL

To my beloved parents Delphine and Didier,
and my dear sisters Clémence and Maguelonne...

Acknowledgment

"A long time ago in a galaxy far, far away...." Star Wars: Episode IV – A New Hope, George Lucas, 1977.

Cette thèse de doctorat a débuté le 18 janvier 2016 et s'est déroulée au CEMEF MINES-Paristech, laboratoire dirigé par le Pr. Élisabeth MASSONI, Directrice du CEMEF, et Mr. Patrick COELS, Directeur Administratif. Je les remercie tous les deux pour leur accueil et mon intégration au sein du laboratoire.

Ce doctorat a été encadré par le Dr. Youssef MESRI et le Pr. Elie HACHEM. Je tiens d'abord à adresser mes remerciements à Youssef qui a été mon mentor pendant ces trois années. Je le remercie pour la transmission de son savoir, sa pédagogie, sa patience et sa disponibilité. Il a été essentiel dans ma formation de chercheur et a toujours fait en sorte que j'en tire le maximum professionnellement. Il a surtout été d'un soutien sans faille, m'a toujours prêté une oreille très attentive. Cela été déterminant dans la réussite de mon projet de thèse. Je remercie également chaleureusement Elie pour les connaissances précieuses qu'il m'a transmises, son expertise et son énergie. Je le remercie pour ces 100 idées par jours, et qu'il en soit sûr, j'aurais adoré les mettre toutes en œuvre si mes journées avaient dépassé 24h. Je tiens particulièrement à le remercier de la confiance qu'il m'a accordée pour mener à bien ce projet, d'avoir toujours cru en moi, et cela depuis le début de ce projet.

La thèse a été entreprise en partenariat CIFRE avec Safran Aircraft Engines (SAE) et je tiens à remercier très chaleureusement Juan-Carlos LARROYA-HUGUET, mon responsable industriel chez SAE. Je le remercie pour les connaissances en turbomachine qu'il m'a apporté, pour son expertise, et pour sa rigueur scientifique que j'ai essayé au maximum de reproduire. Je le remercie enfin pour son écoute, sa disponibilité et pour sa très grande bienveillance à mon égard. Je remercie également toutes les personnes de l'unité YHMT de SAE pour leur accueil lors de mes visites dans l'entreprise. Également, je souhaite remercier tous les partenaires de recherche du groupe Safran pour leur indulgence et la pertinence de nos échanges lors des rencontres que nous avons pu avoir. Je tiens à remercier particulièrement le Dr. Pierre AILLAUD pour m'avoir transmis ses données et pour les échanges que nous avons eus sur le refroidissement des aubes de turbines.

Je souhaite également rendre hommage à tous les auteurs que j'ai pu lire pendant ces trois ans de thèse et notamment, à ceux cités dans ce manuscrit. Je les remercie pour la transmission de leur savoir et les idées que m'ont inspiré leur travaux. Je tiens en particulier à remercier ceux qui font l'effort et prennent le temps d'expliquer les concepts les plus difficiles dans le but de transmettre cette science aux générations futures.

Également, j'adresse ma très grande gratitude à Alexandre BOILLEY qui m'a tout appris de Cimlib-CFD et pour ses travaux qui m'ont permis de gagner un temps non-négligeable en début de thèse. J'adresse aussi mes remerciements à Romain KLEIN qui m'a beaucoup apporté par sa connaissance des systèmes de calculs. Au-delà de ces remerciements professionnels, je souhaite les remercier tous les deux pour les liens amicaux que nous avons tissé pendant ces trois ans et qui m'ont apporté de vrais moments de détentes au Café de Sophia et dans les restaurants de la région. Merci à vous deux et j'espère vous revoir très vite !

Je souhaite également remercier Yuekun YANG, mon stagiaire de Shanghai Jia Tong University pour son très bon travail sur ANSYS CFX et ANSYS Fluent et pour les résultats qu'il m'a transmis et que j'ai pu exploiter dans ce travail de thèse. Puisse-t-il continuer dans cette voie !

Je remercie également Sélim KRARIA pour son aide précieuse dédiée à la compilation de Cimlib-CFD. Je le remercie pour sa disponibilité et le temps qu'il m'a accordé pour m'aider à corriger mes nombreuses erreurs de compilation. Je remercie également Carole THORIN pour son aide sur le Cluster Intel et Hallen BEN AISSA pour l'installation des softs sur ma machine. Merci également à Florence MORCAMP pour son aide et pour sa bonne humeur

Je tiens bien sûr à remercier mes amis du laboratoire qui ont partagé, au quotidien, cette aventure avec moi. Je remercie Ghalia qui a été la première personne à m'accueillir au CEMEF. Je remercie Shitij, Gabriel, Wafa, Danaé, Lucas, Carlos, Victor, Luis, Ramy et Charles pour leurs sourires et nos discussions de haut vol entre midi et deux. Je tiens particulièrement à remercier Chachou pour sa force, sa bienveillance et l'amitié sincère qu'elle m'a manifesté pendant ces trois ans, puisse-t-elle durer ! Je remercie également Robin pour ces (trop rares) soirées jeux, son humour et pour l'amitié que nous avons partagé pendant ces trois ans.

Cette thèse de doctorat n'aura évidemment pas été qu'une expérience professionnelle. C'est l'aboutissement de 9 années d'études, faites liens forts partagés avec des gens véritablement formidables et cela pas uniquement au sein du monde professionnel. Pour cette raison, plusieurs personnes se doivent d'apparaître dans ce manuscrit.

Je tiens d'abord à remercier mes très chers parents, Delphine et Didier, sans qui rien de tout cela n'aurait été possible. Je les remercie infiniment pour leur soutien moral, affectif et financier pendant ces longues années d'études entre la France et l'Angleterre. Je remercie bien sûr mon père qui a été mon premier coach pendant ces années et l'unique personne qui n'a jamais douté de mon succès. Son implication dans ma vie a toujours été très réconfortante et m'a permis de passer certains caps importants aux moments décisifs de ma vie étudiante. Merci Papa de m'avoir toujours encouragé à rester *« calme et félin, souple et manœuvrier...* ». Je remercie également ma mère qui a toujours su me faire voir les choses sous un autre angle: le sien. Son sens de la famille et son sens du « bien vivre » m'ont toujours permis de prendre beaucoup de recul et m'ont apporté énormément de bonheur et de sérénité pendant ces études. En particulier, son soutien affectif et son engagement au plus près de mes projets a été essentiel et je ne la remercierais jamais assez pour cela.

Je souhaite également remercier mes deux sœurs chéries Clémence et Maguelonne. Merci à ma grande sœur Clémence pour sa fidélité et son soutien inconditionnel, aussi bien sur le plan professionnel que sur le plan personnel, et cela alors même que des centaines de kilomètres nous séparaient. Merci d'avoir été mon premier interlocuteur à plusieurs reprises et souvent celui qui me comprenait le mieux. Merci à son mari Naïm qui prend soin d'elle à présent ! À ma petite sœur Maguelonne, milles mercis pour sa fraîcheur, son énergie et sa grande maturité lorsqu'elle donne des conseils à son grand frère. Merci à elle d'avoir été là pour moi et d'avoir répondu au téléphone à toute heure du jour et de la nuit... À nos week-ends entre frangins à Antibes, à Amsterdam et aux futurs à Paris ou ailleurs !

Je tiens également à remercier du fond du cœur AL pour m'avoir accompagné pendant ces deux belles années. Elle a été l'une des rares personnes à vivre cette aventure de l'intérieur. Son soutien dans les moments les plus difficiles de cette thèse a été essentiel et je n'en serais certainement pas là aujourd'hui si elle n'avait pas été là. À l'ensemble de nos souvenirs tous les deux: nos ballades à moto dans l'arrière-pays, nos apéros au coucher du soleil, nos dimanches au calme et nos vacances qui ont été des bouffées d'oxygène dont je n'aurais simplement pas pu me passer.

Je remercie également mon team du passé, du présent et du futur, bref mon équipe solide et éternelle, j'ai nommé ici : Anto, Binouch, El Gablo, Greg, Pipo, Six-F, Dr. Spirz et La Vipère pour leur soutien pendant ces années d'études, sans doute beaucoup trop longues à leurs goûts. Je les remercie mille fois pour ces week-ends à Paris qui m'ont permis de me retrouver lors de mes retours au bercail. Vous êtes des frères. Cela fait 10 ans qu'on avance ensemble et j'ai hâte de voir ce que les 10 prochaines années nous réservent... Merci à Anto pour cette amitié qui dure depuis si longtemps et pour son accueil toujours aussi sincère et chaleureux lors de mes séjours à la capitale. Je me permets ici aussi une mention spéciale à Greg, qui a eu le bon sens de me citer Booba quand je lui ai annoncé que j'étais pris en thèse de doctorat à l'autre bout de la France : *« All in, carré d'as »*. Je le remercie pour les moments qu'on a passé tous les deux dans le sud, à l'Épi Beach ou dans les montagnes, qui m'apportent aujourd'hui de très bons souvenirs de ma vie là-bas.

Je souhaite, bien évidemment, remercier ici la tribu des Zoulous de l'Ecole Centrale de Nantes et je ne prendrais pas le risque de n'en nommer que quelque uns dans ce manuscrit car sachez bien qu'ils y sont tous pour quelque chose. À nos week-ends printaniers à Suzette qui m'ont permis de garder contact et de vous revoir tous régulièrement, dans une ambiance de franche camaraderie. À nos week-end chez le Sweeny à la Ciotat, pas loin d'ici, qui m'ont permis de décompresser à seulement quelques kilomètres de chez moi. J'espère vous revoir tous très bientôt à Paris ou ailleurs. Amour, joie, paix et prospérité à tous les Zoulous aux quatre coins de la planète.

Je tiens également à remercier ici la famille Bazile/Guinot : Coulou, Alain, Isabelle, Hélène, Damien, Maxence, Dianou, Victor, mon filleul Gustave, le Toon, Laëtitia et Glenn pour les moments que nous partageons depuis nos naissances respectives. Mention spéciale pour mon Toon qui m'a appelé régulièrement pendant ces trois ans pour s'assurer que *« tout allait bien dans ma life »*. À nos diners en tête à tête en plein cœur de Paris à refaire le monde et nos vies. Je remercie également la famille Bardin : ma grand-mère Nicole, Nicolas, Caroline, Eugénie, Jeanne, Lucie, Joséphine et ma tantine Olivia pour nos Noël toujours plus féériques à la Cherbonnais qui nous font tant de bien mais aussi pour nos moments de retrouvailles à la Gâtine. Je tiens aussi à remercier et à saluer chaleureusement la tribu Lopin pour leur bonne humeur infaillible avec une mention spéciale à mon parrain Jérôme qui a suivi de près la transition de cette thèse avec la suite de mon parcours. Je remercie également ma marraine Charlotte pour ce super diner au Belambra !

Je souhaite également remercier certaines personnes un peu plus éloignés de cette aventure mais tout de même essentielles à mon bien-être au travail et à mon inspiration créative. Je remercie George Lucas pour l'ensemble de son œuvre exceptionnelle, intemporelle et si inspirante qu'elle a été un des principaux drivers de ces longues années d'études. Je remercie également John Williams, Hans Zimmer, Steve Jablonsky, Dan Ar Braz, James Horner, Ramin Djawadi, Era, Lunasa, Alexandre Desplat, Howard Shore, Alan Stivell, Bear McCreary, Alan Silvestri et Nicholas Hooper pour leurs œuvres respectives qui m'ont accompagné quotidiennement pendant ces 9 années de travail.

Je remercie enfin toute autre personne qui aurait participé, de près ou de loin, à la réussite de ce projet de thèse et, plus largement, à la réussite de mes études. Un long chapitre vient de trouver sa fin alors qu'un nouveau s'ouvre déjà et l'Avenir, avec vous tous, s'annonce très prometteur...

"Do not go gentle into that good night. Rage, rage against the dying of the light." Dylan Thomas, 1951.

Contents

Chapter 1

Introduction

Contents

1.1 Industrial context

By 2030, considering the progress of High Performance Computing (HPC), aerospace manufacturers like Safran Aircraft Engines (SAE), hope to be able to simulate a whole aircraft engine, at full scale, using Computational Fluid Dynamic (CFD). Indeed, in the next 20 years, their objective is to replicate all the physical phenomenon that occur inside an aircraft engine in order to certify it without using expensive real-life tests rigs. However, some parts of the engine are still challenging to model with existing codes. In particular, we can identify among them: (i) the rotating parts, (ii) the reactive flows in the combustion chamber and (iii) the cooling systems. These three parts are presented in Figure 1.1. For the rotating parts, the difficulty is to be able to move the mesh at high speed inside another domain using efficient moving mesh methods. For reactive flows however, high fidelity chemical models are needed to describe the gas reactions inside the combustion chamber and to ensure a strong coupling with the flow dynamics.

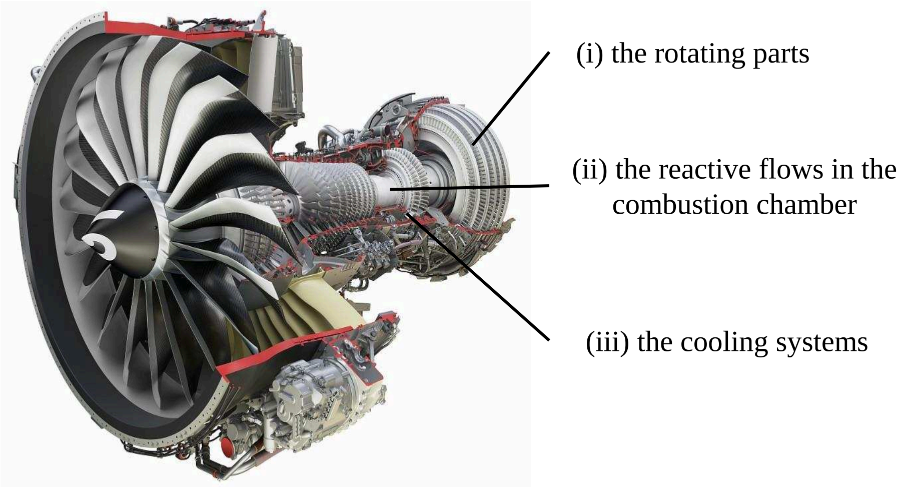

(i) the rotating parts
(ii) the reactive flows in the combustion chamber
(iii) the cooling systems

Figure 1.1 – Slice view of Safran Aircraft Engines LEAP 1B
Credit: Safran Aircraft Engines.

1.1.1 Generalities on the cooling of aircraft engines

Concerning the cooling systems, their design plays a critical role in the life span of the aircraft engine components. As mentioned by Hetmańczyk et al. in [1], the gas turbine engines operate in one of the harshest environments. Indeed, their components are subjected to severe mechanical loads, high temperatures, corrosion and erosion. These difficult conditions often cause considerable damages on the engine components, that can even lead to cracking on one of the engine parts. A way to improve the parts resistance is by studying the micro-structure of the material itself. To do so, mechanical engineers like in [1] have tried to find advanced alloys materials capable of dealing with these harsh conditions. Another way consists in improving the efficiency of the cooling systems in order to keep the material in an acceptable range of temperature to avoid any weakening of the part caused by meeting with hot gases.

However, the efficiency of the cooling systems has been increasingly endangered over the past decades. Indeed, in the last decades, we have observed an increase of the turbofans By-Pass Ratio (BPR) which is defined by the following formula:

$$\mathrm{BPR} = \frac{\dot{m}_{secondary}}{\dot{m}_{primary}} \tag{1.1}$$

where $\dot{m}_{primary}$ and $\dot{m}_{secondary}$ are the primary and secondary mass flow rates. For the reader understanding, the primary and the secondary flows are represented on the schematic illustration given in Figure 1.2.

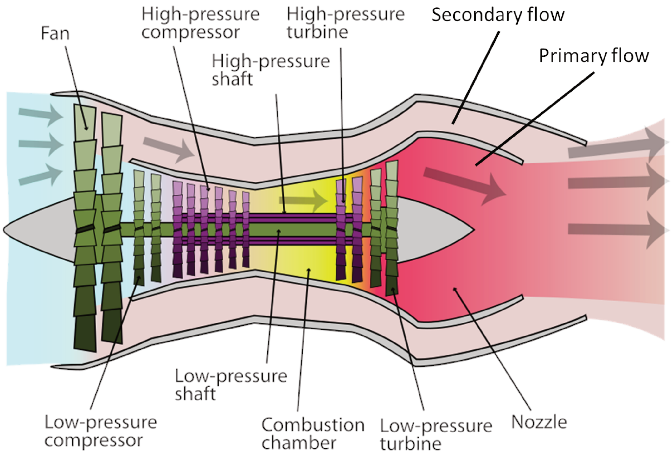


Figure 1.2 – Major components of a turbofan. Primary and secondary flows.
Credit: Wikipedia.

As shown by Verseux et al. in [2], the increase of the BPR induces a decrease of the Fan Pressure Ratio (FPR). Some representative FPR of aircraft engines composing the Airbus fleet are given in Figure 1.3. The diminishing of the FPR implies that the overall pressure of the turbofan's secondary flow decreases. The issue is that the turbofan's secondary flow is the major source of bleed air of the cooling systems, and this pressure reduction can deteriorate significantly the cooling performances. Over the years, the challenges around the design of aircraft engine cooling systems have thus been increasingly important.

As an example, Airbus placed a particular attention to the cooling of the engine nacelle in the New Engine Option (NEO) program. Among other reasons, they chose the LEAP engine from SAE to equip the new Airbus A320 NEO thanks to the good cooling performances of the latter.

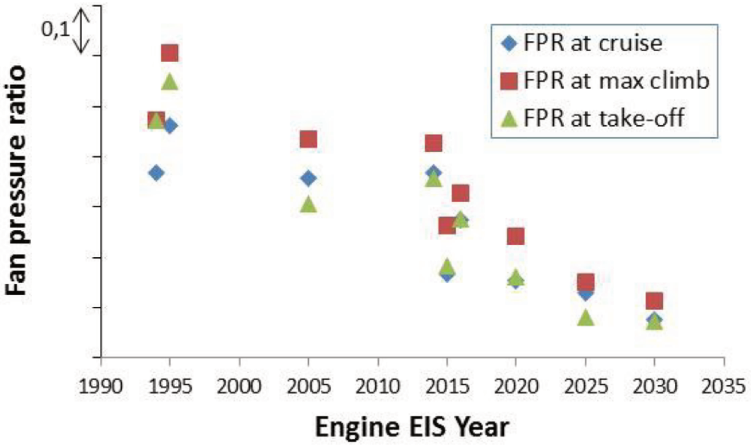


Figure 1.3 – FPR versus engine's Entry Into Service (EIS) year for the Airbus fleet [2]

1.1.2 High pressure turbine blade cooling

In this thesis, the focus is put on the cooling systems of the turbine blades (or vanes) located on the stator part of the high pressure turbine. They are represented on the illustration from Schwerdt et al. in Figure 1.4. As show in Figure 1.2, the high pressure turbine is located right after the combustion chamber. Therefore, the gases arriving on its stator blades are at very high temperature. In particular, the fusion temperature of the alloys constituting the stator blades is below the gases temperature coming from the combustion chamber. Indeed, as mentioned by Terzis et al. in [3], the estimated Turbine Entry Temperature (TET) has been increasingly important over the last decades and went progressively above the fusion temperature of most alloys. Consequently, it is of first importance to cool down the turbine blades in order to ensure that the material will not melt under the hot gases coming from the combustion chamber.

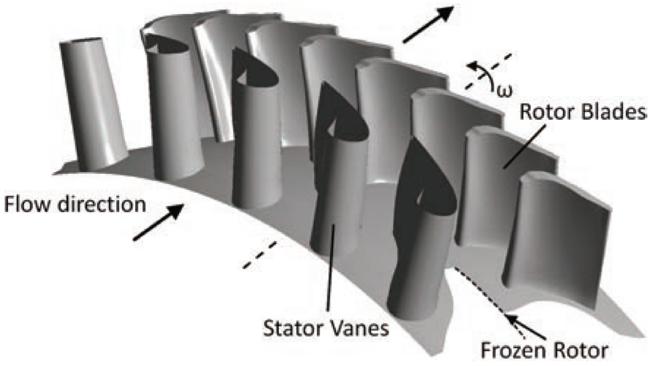


Figure 1.4 – Stator blades (or vanes) of the high pressure turbine [4]

To do so, aircraft manufacturers bring into play two main principles of cooling: (i) the film cooling and (ii) the internal impingement jet cooling. These two types of cooling are presented in Figure 1.5. We observe that, from a single network of pipes inside the turbine blade, the cooling is realized thanks to different physical phenomenon regarding the location of study. On the blade extrados and intrados, the film cooling is the main mean of cooling. Inside the turbine blade however, the internal impingement jet cooling is used to cool down the inner part of the blade.

In this study, we focus our interest on the internal impingement jet cooling. For this case, the cooling is realized thanks to the injection of a cooled airflow inside the blade. Indeed, as shown in Figure 1.6, a cavity is scooped inside the blade and pierced with several holes that make it possible for the airflow to impact the internal wall of the blade in order to cool it down by forced convection. The objective is then to know accurately the heat transfer coefficient distribution over the inner part of the turbine blade in order to predict the cooling performances.

The strategy defined by SAE to treat this complex industrial case is by using three incremental difficulties. The first one concerns the study of a single jet, impinging normally on a hot flat plate, representative of the extrados/intrados part of the turbine blade. The second case is a raw of jets, impinging normally on a concave plate, representing the leading edge of the blade. Finally, the last case is a complete geometry of the turbine vane with a network of jets interacting with each other.

1.1.3 Impingement jet cooling in cross-flow

Following this strategy, the first geometry studied in this PhD project was a single jet, impinging normally on a hot flat plate, in a cross-flow. This geometry is in fact representative of the extrados/intrados area of the blade. Indeed, if we now isolate a unique jet from the extrados (or in-

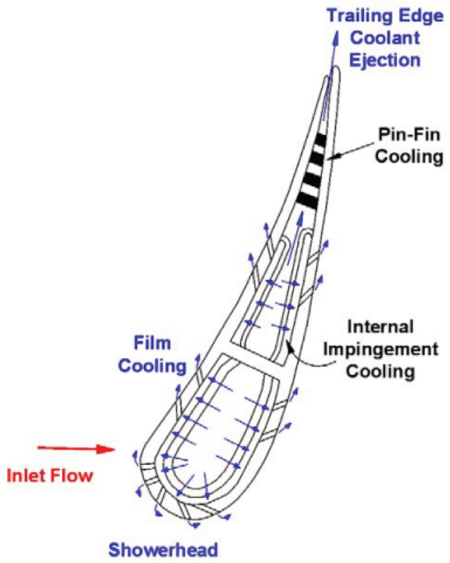


Figure 1.5 – Slice view of a turbine blade equipped with different cooling systems [5, 6]

.

trados) area, the situation can be summarized by a turbulent jet, coming from the blade's cooling area and crossed by the evacuation airflow as described on the right side of Figure 1.6.

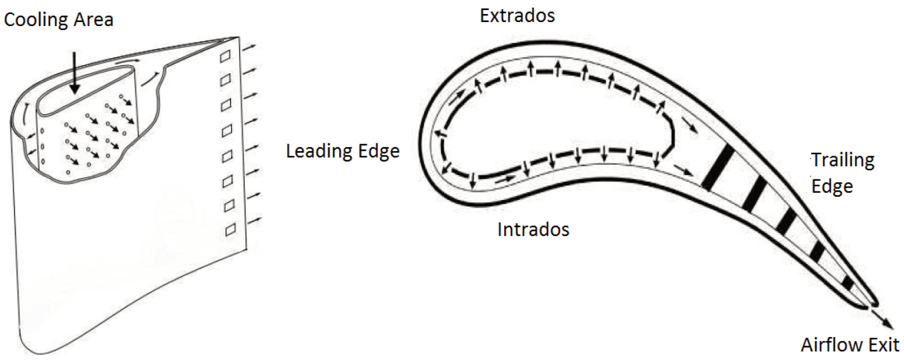


Figure 1.6 – Schematic view of the turbine blade cooled by internal impingement jets [7]

.

This configuration was studied experimentally by Thibault et al. in [7] in order to produce an experimental database for future numerical simulations. The author proposes in Figure 1.7 a schematic view of the jet in cross-flow. The primary flow is the flow coming from the cooling area. It circulates inside the turbine vane before going through a hole, pierced in the injection plate. Then, the primary flow impacts the plate in order to cool it down. The gap area between the injection and the impact plate starts heating up by the effects of the hot plate. Thus, a secondary flow crossing the primary flow evacuates the heat from the gap area in the X axis direction. The goal of this aerothermal problem is then to obtain the heat transfer coefficient distribution on the impact plate.

This case is extremely strategic for SAE. Indeed, many attempts to solve this complex aerothermal were made in the past. Huynh et al. in [8] tried to solve this case with the commercial software ANSYS Fluent using Reynolds-Averaged Navier-Stokes (RANS) / Unsteady Reynolds-Averaged Navier-Stokes (URANS) methods to test different turbulence models like the $k-\epsilon$, $k-\omega$ or $v^2 f$

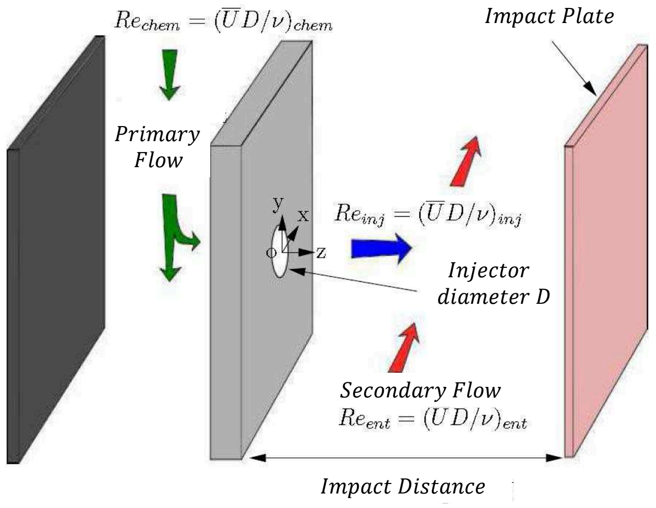

$Re_{chem} = (\overline{U}D/\nu)_{chem}$
Impact Plate
Primary Flow
y
x
o
z
$Re_{inj} = (\overline{U}D/\nu)_{inj}$
Injector diameter D
Secondary Flow
$Re_{ent} = (UD/\nu)_{ent}$
Impact Distance

Figure 1.7 – Schematic view of the impingement jet in cross-flow [7]

.

models. Other numerical studies like in [9] were made at the Office National d'Etudes et de Recherche Aérospatiales (ONERA) using hybrid numerical methods like the Zonal Detached Eddy Simulation (ZDES) with parietal laws, computed with the CEDRE code, coupled with the Navier-Stokes solver CHARME. For now, these numerical studies were not able to predict accurately the heat transfer coefficient distribution on the impact plate. Indeed, in all of these research works, an over estimation of the heat transfer coefficient distribution was found.

From these previous numerical studies, and from the preliminary results we obtained on this geometry, some critical difficulties were reported to the research network of SAE. These critical difficulties were first related to the aerodynamic boundary conditions of the problem. Indeed, we noted that the velocity profiles at the geometry's inlet and the turbulence rate of the primary flow had not been measured by the experimental studies. For this reason, it was difficult for computer scientists to know a priori the right inlet boundary condition to put on the geometry. Furthermore, the pressure differential between the primary and the secondary flows was not clearly defined in the experimental setup and thus, had to be assumed. After some numerical experiments made during this PhD work, we managed to show that this lack of information had a damaging effect on the heat transfer results. Finally, these difficulties were emphasized by the lack of references on this particular case in the literature.

The goal of this PhD project is to address the challenges around the impingement jet cooling and to find in what measure we are able to tackle them. To do so, we were advised by SAE to focus our work on a more simple case, closely linked to the jet in cross-flow. Indeed, from our preliminary analysis on the jet in cross-flow, the challenges of the cooling by impinging jet are mostly related to the impinging jet itself, and our ability to capture the parietal heat exchanges in the boundary layer formed by the impact plate. Furthermore, to our knowledge, the impingement jet cooling has never been studied using an eulerian finite element framework. It was thus necessary to study more in details a benchmark configuration of the latter in order to validate the finite element framework on this type of aerothermal problems. This benchmark configuration study is the object of Chapter 2 and it is, in particular, the common thread running through this PhD work.

1.2 State of the art

Over the last years, CFD in aerospace industry has reached a technological cross-road for several reasons. Slotnick et al. from National Aeronautics and Space Administration (NASA) in [10] propose to summarized these reasons in a technological road map given in Figure 1.8. Indeed, for several years now, despite the fact that RANS and Hybrid RANS/Large Eddy Simulation (LES) methods with improved Reynolds Stress Transport (RST) methods have reached high Technology Readiness Level (TRL), computer scientists have observed that they are still unable to predict with high accuracy the aerodynamic of turbulent flows in a large number of cases. In particular, as seen in the last section, it brings issues in aerothermal simulations where the heat exchanges are majorly convective and occurs in turbulent boundary layers. The difficulty met by these methods is to predict the turbulent flow separation in the boundary layers at flight Reynolds numbers i.e. at Reynolds numbers above 10^5. The technology demonstration of these methods is indeed due in 2020. After this point, LES methods will need to demonstrate better TRL in order to allow computer scientists to simulate unsteady separated flows in complex 3D geometries like the ones met in rotating turbomachinery for example.

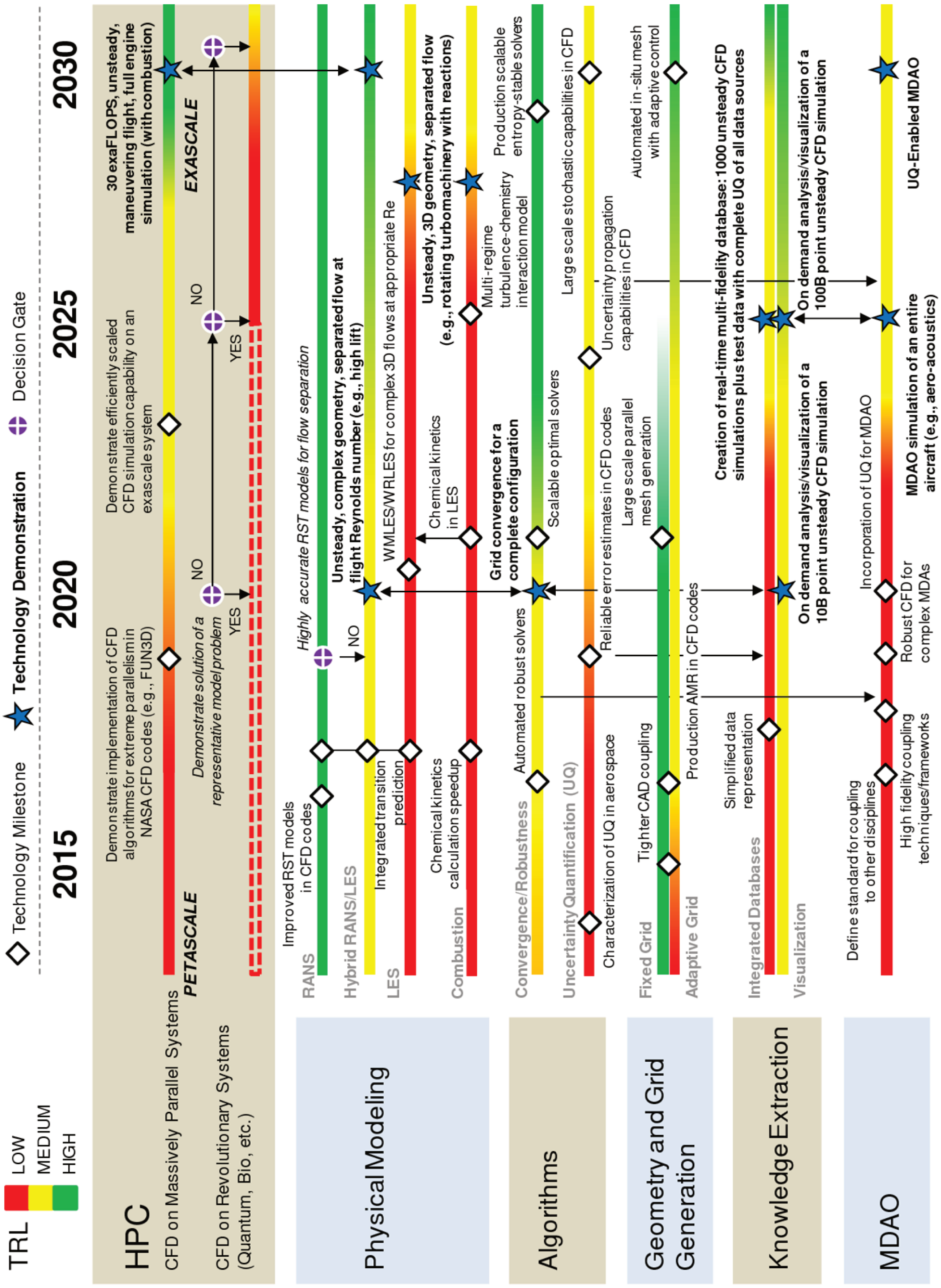

TRL
LOW
MEDIUM
HIGH
Technology Milestone
Technology Demonstration
Decision Gate
2015
2020
2025
2030
HPC
CFD on Massively Parallel Systems
CFD on Revolutionary Systems (Quantum, Bio, etc.)
PETASCALE
EXASCALE
Demonstrate implementation of CFD algorithms for extreme parallelism in NASA CFD codes (e.g., FUN3D)
Demonstrate efficiently scaled CFD simulation capability on an exascale system
30 exaFLOPS, unsteady, maneuvering flight, full engine simulation (with combustion)
Demonstrate solution of a representative model problem
YES
NO
Physical Modeling
RANS
Hybrid RANS/LES
LES
Combustion
Improved RST models in CFD codes
Highly accurate RST models for flow separation
Integrated transition prediction
Unsteady, complex geometry, separated flow at flight Reynolds number (e.g., high lift)
WMLES/WRLES for complex 3D flows at appropriate Re
Unsteady, 3D geometry, separated flow (e.g., rotating turbomachinery with reactions)
Chemical kinetics calculation speedup
Chemical kinetics in LES
Multi-regime turbulence-chemistry interaction model
Production scalable entropy-stable solvers
Algorithms
Convergence/Robustness
Uncertainty Quantification (UQ)
Automated robust solvers
Grid convergence for a complete configuration
Scalable optimal solvers
Characterization of UQ in aerospace
Reliable error estimates in CFD codes
Uncertainty propagation capabilities in CFD
Large scale stochastic capabilities in CFD
Geometry and Grid Generation
Fixed Grid
Adaptive Grid
Tighter CAD coupling
Production AMR in CFD codes
Large scale parallel mesh generation
Automated in-situ mesh with adaptive control
Knowledge Extraction
Integrated Databases
Visualization
Simplified data representation
Creation of real-time multi-fidelity database: 1000 unsteady CFD simulations plus test data with complete UQ of all data sources
On demand analysis/visualization of a 10B point unsteady CFD simulation
On demand analysis/visualization of a 100B point unsteady CFD simulation
MDAO
Define standard for coupling to other disciplines
High fidelity coupling techniques/frameworks
Robust CFD for complex MDAs
Incorporation of UQ for MDAO
MDAO simulation of an entire aircraft (e.g., aero-acoustics)
UQ-Enabled MDAO

Figure 1.8 – NASA CFD technology development road map for aerospace industry [10]

Along these physical modeling considerations, research efforts are also needed at a software level. Indeed, according to the technology road map, the convergence and robustness of CFD solvers has not yet reached high TRL. The grid convergence is still difficult to achieve in a number of turbulent cases where prohibitive computational resources are needed. For this reason, research work on the scalability and efficiency of solvers needs to be continuously updated, and foundational mathematical research in highly scalable linear and non linear systems is still required. In particular, to improve the efficiency of numerical solvers, new reliable error estimates need to be designed, tested and validated on realistic industrial cases. Improvements on error analysis are also required to asses for the propagation of uncertainties during simulations. In particular, discretization error estimation is a key ingredient for the realization of an adaptive solution process.

Indeed, closely linked to discretization error estimates, Adaptive Mesh Refinement (AMR) methods have yet to prove their capabilities in industrial CFD codes. For a few years now, aerospace industry has welcomed these new techniques allowing to automatically adapt meshes according to the simulations needs. The AMR strategies offer indeed the potential of a superior accuracy at a reduced cost. However, these methods were, until recently, mostly applied to academic cases where limited computational resources were needed. In fact, they were not widely spread due to code robustness and software complexity issues. For many flows involving difficult flow physics and complex geometries, many of the current AMR techniques are not strong enough to ensure a robust solver convergence. For this reason, NASA predicts an entry in industrial production of these adaptive techniques not until 2020. Before that, these techniques will need to demonstrate their capabilities on realistic industrial cases where large computational resources are involved.

In fact, the fast development of the HPC at the hardware level keeps pushing computer scientists to readjust their numerical tools, in order to improve the performances of their CFD codes on new emerging hardware architectures. According to the NASA report, considering the actual evolution of technologies, computer scientists predict that the future hardware systems will reach a computational peak capacity of 30 Exa FLoating-point Operations Per Second (FLOPS) in 2030. To reach this performance, it will be necessary to operate a complete change of paradigm in the numerical simulation. This change of paradigm will require a complete restructuring of the hardware architectures but also an effort from the numerical scientists to adapt their numerical methods to the new computational systems.

Indeed, until today, the majority of numerical method developers did not take into account these hardware considerations. It is therefore very difficult for today's computer scientists to capitalize as much as possible on the new emerging hardware architectures. An algorithmic research work effort is thus needed to continuously improve the scalability of numerical methods. This notion is called the "co-design". The co-design allows developing numerical methods in close consultation with the progress of HPC. In particular, it allows taking into account a high level of parallelism in hardware systems during the development of numerical methods.

These considerations have also been noticed by the members of the ExaFLOW project in [11, 12]. In this project, they summarize the scientific locks that CFD has to overcome to reach the *exascale* computation. They are stated as follows:

1. error control and adaptive mesh refinement in complex computational domains,
2. resilience and fault tolerance in complex simulations,
3. heterogeneous modeling,
4. evaluation of energy efficiency in solver design,

5. parallel input/output and in-situ compression for extreme data.

1.3 Contribution of the thesis

The goal of this PhD thesis is then to bring a scientific contribution to this research framework. Indeed, we propose in this work the development of a variational adaptive finite element method in a massively parallel computational framework allowing to improve the aerothermal turbulent simulations related to the cooling of aircraft engines. More precisely, our goal is to develop a new multiscale mesh adaptation technique, adapted to the resolution of the highly convective aerothermal problems in turbulent flows.

To do so, we first propose to study a complex industrial problem that requires heterogeneous modeling taking into account: (i) the part's thermal history, (ii) the unsteady and turbulent nature of the flow, (iii) a complex three dimensional (3D) geometry, and above all, (iv) the multiscale dominant aspects of the flow. In fact, we propose to use a novel set of CFD tools to solve the aerothermal problem of the turbine vane cooling. More precisely, the research effort is put on a benchmark configuration of the single impingement jet cooling. To our knowledge, this is the first attempt to solve the impingement jet cooling problem using an adaptive eulerian finite element framework.

To solve this aerothermal problem, we solve two different decoupled Partial-Differential Equation (PDE) systems. The first PDE system is dedicated to the solving of the unsteady incompressible Navier-Stokes equations. It uses a stabilized finite element methods called the Variational Multi-Scale (VMS) formulation under the assumption of a convection dominated regime. The second PDE system is dedicated to the solving of the convection-diffusion equation for the temperature. It takes, as an input and from the previous resolution, the resulting velocity in the convection term. We refer to the use of the well known Streamline Upwind Petrov/Galerkin (SUPG) scheme introduced in [13] to stabilize the standard Galerkin formulation. Furthermore, as we solve decoupled aerothermal problems, additional numerical models are needed in order to achieve aerothermal coupling in the simulation. In this work, we use two different numerical models simultaneously: (i) the Sutherland law and (ii) the Smagorinsky Model (SM). These numerical models are tested and validated on the benchmark case of a 3D ventilated cavity.

Furthermore, a new hierarchical control of errors is proposed via the development of a new multiscale mesh adaptation technique based on recently developed subscales error estimators. In this thesis, we compute the subscales error estimator using two different methods. The first one uses stabilizing parameters derived from the VMS analysis and local norms defined on the elements. The second method uses a linear combination of bubble functions to establish a pointwise computation of the error. The first contribution of this work is then to propose a new *isotropic* mesh adaptation technique based on the previous subscales error estimator. To do so, we define a new isotropic metric tensor $\mathcal{H}_{iso}$ and we solve an optimization problem under the constraint of a fixed number of elements. Until now, the VMS error estimators were only derived for isotropic mesh adaptation. In this work, we also propose to use the sub-mesh scale information for anisotropic mesh adaptation.

The second contribution is therefore to combine both the coarse scales interpolation error indicator and the subscales error estimator for *anisotropic* mesh adaptation. To do so, we derive a new anisotropic metric tensor $\mathcal{H}^{new}_{aniso}$ that allows taking into account the anisotropic variations of the solution on the mesh and also relies upon the sub-grid information of the solution. This new anisotropic mesh adaptation technique takes into account the finite element VMS framework used for the numerical resolution of the convection-diffusion equation. In fact, the VMS approach allows quantifying the a posteriori subscales errors that can be used as weight to enrich the anisotropic error indicator of the conventional Hessian based approach. Unlike existing anisotropic a posteriori error estimators, this hybrid approach takes advantages from both a priori and a posteriori methods.

Finally, we propose a description of the parallel software capabilities of our finite element library Cimlib-CFD. First of all, we present the Finite Element (FE) framework, the mesh partitioning principles and the load balancing algorithms. Then, the algorithm behind parallel mesh adaptation is introduced as well as important notions on the particular treatment of the partitions' interfaces. A mention of the parallel visualization tool is also proposed allowing to complete the parallel computational framework developed in this work. Second of all, we propose a presentation of the two hardware systems used to develop numerical tools and to produce numerical results during the thesis. The first one is the lab's cluster, called Cluster Intel, allowing the development of numerical methods. The second one is the GENCI Occigen II supercomputer on which we executed Cimlib-CFD for the first time during this PhD work. This system allows producing numerical results using massively parallel computations. Taking into account the characteristics of the super-computer, a fault tolerant checkpoint-restart procedure is developed. We then propose application cases for parallel computations. The first one is dedicated to a strong scalability analysis of our finite element code on a well known benchmark. Then, parallel mesh adaptation is tested on the two hardware systems for the case of the single impingement jet cooling. Finally, we present a more complex but realistic industrial case that concerns the cooling of a complete turbine vane composed by 39 holes.

1.4 Author's contributions during the PhD

Refereed Journal Articles

1. Bazile, A., Hachem, E., Larroya-Huguet, J. C., & Mesri, Y., *Variational Multiscale error estimator for anisotropic adaptive fluid mechanic simulations: Application to convection–diffusion problems.* Computer Methods in Applied Mechanics and Engineering, Vol. 331, pp 94-115, 2018.

2. Bazile, A., Mesri, Y., Larroya-Huguet, J. C., & Hachem, E., *Aerothermal Impingement Jet Flow Simulations using Anisotropic Multiscale Mesh Adaptation.* In 2018 Fluid Dynamics Conference, 2898, AIAA AVIATION Forum, 2018.

3. Mesri, Y., Bazile, A., Larroya-Huguet, J., & Hachem, E., *Parallel and adaptive VMS finite elements formulation for aerothermal problems.* Computer & Fluids, Vol 173, pp 42-50, 2018.

Communications

1. Bazile, A., Mesri, Y., Larroya-Huguet, J., & Hachem, E., *Formulation adaptative variationnelle et calcul massivement parallèle pour l'aérothermique industrielle avec Safran Aircraft Engines.*, Research & Technology Day 2018, Safran Aircraft Engines, Villaroche, France, 2018.

2. Bazile, A., Mesri, Y., Larroya-Huguet, J., & Hachem, E., *Turbine blade cooling simulations using parallel anisotropic multiscale mesh adaptation*, Journée Des Doctorants Safran Mécaniques des Fluides, Safran Tech, Magny-les-Hameaux, France, 2018.

3. Bazile, A., Mesri, Y., Larroya-Huguet, J., & Hachem, E., *LES multi-physique avec maillage adaptatif et refroidissement par impact,* Journée Des Doctorants Safran Mécaniques des Fluides, Safran Helicopter Engines, Bordes, France, 2017.

4. Bazile, A., Mesri, Y., Larroya-Huguet, J., & Hachem, E., *Formulation adaptative variationnelle et calcul massivement parallèle pour l'aérothermique industrielle avec Safran Aircraft Engines.* CSMA 2017. 13ème Colloque National en Calcul des Structures, 2017.

5. Bazile, A., Hachem, E., Larroya-Huguet, J. C., & Mesri, Y., *Parallel and adaptive VMS finite elements formulation for aerothermal industry with Safran Aicraft Engines*, 29th International Conference on Parallel Computational Fluid Dynamics, Strathclyde in Glasgow, Scotland, 15-17 May 2017.

6. Mesri, Y., Bazile, A., & Hachem, E., *A predictive load balancing approach for parallel anisotropic adaptive finite element solvers in aerodynamics.* In 19th International Conference on Finite Elements in Flow Problems FEF 2017, 2017.

7. Hachem, E., Billon, L., Boilley, A., Bazile, A., & Mesri, Y., *Anisotropic boundary layer mesh generation for immersed complex geometries.* In 19th International Conference on Finite Elements in Flow Problems FEF 2017, 2017.

8. Mesri, Y., Bazile, A., & Hachem, E., *On dynamic load balancing schemes for adaptive finite element solvers*, VII International Conference on Coupled Problems in Science and Engineering, 2017.

Bachelor's Thesis

1. Yang, Y., Bazile, A., Mesri, Y., Yang, M. & Rao, Y. *Aerothermal Impingement Jet Flow Simulation And Analysis For Aircraft Engines*, Bachelor's Thesis, Shanghai Jiao Tong University, 2018.

Awards

1. Finalist of the Prix de la Médaille Pierre Laffitte 2017

Résumé du chapitre en français

Ce chapitre d'introduction nous permet d'abord de présenter le contexte industriel de cette thèse en partenariat Convention Industrielle de Formation par la REcherche (CIFRE) avec SAE. Nous proposons ici une description générale autour des enjeux du refroidissement des turbomachines notamment via une analyse sur la diminution du FPR. En effet, cette diminution a pour effet de fortement dégrader les performances des systèmes de refroidissement actuels et rend les problématiques autour du refroidissement des moteurs de plus en plus difficiles. Ensuite, certaines considérations sur le refroidissement des aubes de turbines sont présentées. Notamment, il est expliqué que l'augmentation de la température d'entrée des gazes dans la turbine haute pression (TET) peut avoir un fort impact sur la résistance des matériaux qui la compose. Les techniques de refroidissement des aubes sont décrites et plus particulièrement, la technique de refroidissement par impact de jets sur la paroi interne de l'aube. La stratégie à complexité incrémentale de SAE sur ce cas aérothermique est présentée. Il s'agit d'abord (i) de traiter le cas du refroidissement d'un seul jet sur plaque plane, pour ensuite (ii) traiter le cas d'une rangée de jet sur paroi concave, pour enfin (iii) s'intéresser à la géométrie complète de l'aube. Suivant cette stratégie, nous présentons une géométrie préliminaire de jet à écoulement cisaillant amont étudiée en début de thèse. Cette géométrie a été étudiée expérimentalement par d'autres laboratoires mais elle reste aujourd'hui trop complexe à mettre en place pour plusieurs raisons. Dès lors, nous avons décidé de nous ramener à une géométrie benchmark du même type mais plus largement étudiée dans la littérature. Cette géométrie benchmark du jet impactant constitue notamment le fil rouge de ces travaux de thèse.

Nous proposons également dans ce chapitre un état de l'art technologique des méthodes numériques liées à la CFD. En se basant sur un rapport de la NASA, nous mettons en évidence les difficultés des méthodes actuelles, utilisées dans l'industrie, à déterminer avec précision l'aérodynamique des écoulements turbulents. En parallèle de ces difficultés de modélisation physique, certaines considérations softwares sont prises en compte. Elles concernent notamment l'efficacité des solvers numériques et le développement d'estimateurs d'erreur plus performants pour les améliorer. Etroitement lié à cela, l'avancée technologique des méthodes d'adaptation de maillage est présentée ainsi que la notion de co-design. Ces considérations sont reprises par les participants du projet ExaFLOW qui résument parfaitement les objectifs de la CFD dans les années à venir:

1. le contrôle hiérarchique des erreurs et l'adaptation de maillage sur des géométries complexes,

2. la résilience et et la "fault tolerance" des systèmes complexes,

3. l'efficacité des solveurs par des méthodes de Galerkin mixtes, continues ou discontinues et par le choix approprié de préconditionneurs,

4. la modélisation hétérogène,

5. l'évaluation de l'efficacité énergétique dans l'implémentation du solveur,

6. l'implémentation parallèle et la compression des données de calcul.

L'objectif de ces travaux de thèse est donc d'apporter une contribution scientifique à ce contexte numérique. Pour cela, nous proposons de résoudre un cas industriel complexe faisant intervenir une modélisation hétérogène prenant en compte: (i) l'histoire thermique de la pièce, (ii) la nature turbulente et instationnaire de l'écoulement, (iii) une géométrie complexe en 3D et (iv) les aspects multi-échelles de la simulation. De plus, dans ce travail, nous proposons un contrôle hiérarchique des erreurs via le développement d'une nouvelle méthode d'adaptation de maillage multi-échelle basée sur le développement récent d'estimateurs d'erreur à posteriori. Pour cela, nous définissons deux nouveaux tenseurs métriques $\mathcal{H}_{iso}$ et $\mathcal{H}^{new}_{aniso}$ qui permettent l'adaptation isotrope du maillage en se basant sur l'information sous-échelle mais également la combinaison de cette information avec les méthodes d'adaptation de maillage anisotropes conventionnelles basées sur le Hessien. Enfin, ces méthodes numériques sont développées dans un contexte de calcul massivement parallèle que nous nous attachons à présenter dans le dernier chapitre. Notamment, ce dernier chapitre permet de montrer les premiers résultats de Cimlib-CFD sur le supercalculateur national GENCI Occigen II avec la mise en place d'un cas massivement parallèle représentant une aube de turbine complète composées de 39 trous.

Chapter 2

The Single Impingement Jet Cooling

Contents

2.1 Introduction

This chapter is dedicated to the study of a single unconfined isothermal jet impinging normally on a hot flat plate. In the past decades, several experimental and numerical studies like in [14–19] have been made on this aerothermal problem. An example of a smoke visualization from the experimental studies of Cornaro et al. in [16] is given in Figure 2.1. In the past, a large number of different configurations have been studied. In particular, these studies have shown that the configuration characteristics that had the most influence on the plate cooling were: (i) the shape of the jet nozzle (round or square) (ii) the injection Reynolds number Re_{inj} (i.e. the Reynolds number inside the inlet tube) and (iii) the normalized impact distance H/D, with H being the impact distance and D the jet nozzle diameter.

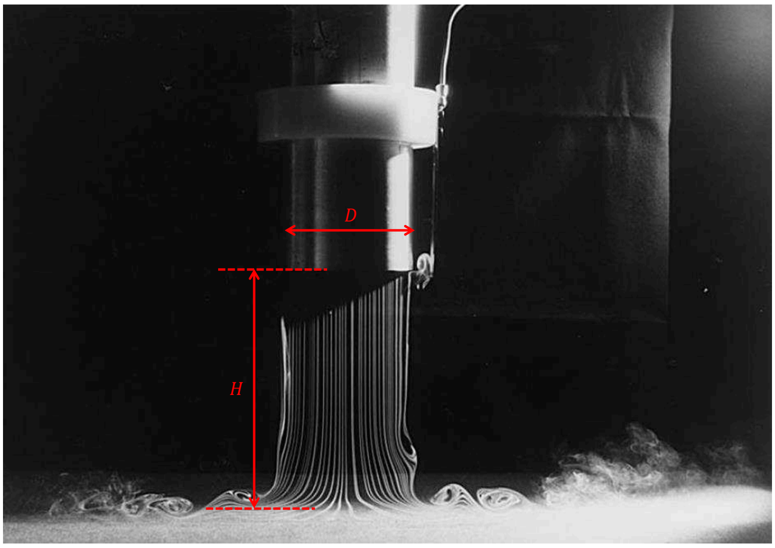


Figure 2.1 – Experimental visualization of a single jet impinging on a flat plate for Re_{inj} = 60 000 and H/D = 2 from Cornaro et al. in [16]

In particular, for an impact distance of H/D = 2 and injection Reynolds numbers above 10 000, it has been noticed by experimental studies that the flow turbulence induced a very specific heat transfer distribution on the plate. In fact, for this particular impact distance, the rebound of the jet primary structures on the plate induces the creation of turbulent secondary vortices that generate a local increase of convective heat transfer on the plate. Because of this local increase, a secondary peak appears in the radial Nusselt number distribution. To better understand this mechanism, the reader can refer to Figure 2.14 that will be presented later in this chapter. The secondary peak is visible in the experimental results from Cooper et al. in Figure 2.2. Our study aims at capturing this specific radial Nusselt number distribution associated with the particular impinging jet configuration of H/D = 2.

In the past few years, several research units have tried to capture this secondary peak using HPC associated with powerful CFD codes. Among them, important work from Tsubokura et al. was made in [17] describing the flow behavior at the injection hole exit. In [20], Grenson et al. proposed a LES simulation of the impingement jet cooling with Reynolds numbers up to 60 000 using the finite-volume solver elsA from ONERA. More recently, Uddin et al. in [21] performed a fully 3D LES simulation in the exact same configuration using the FASTEST code. As well, Dairay et al. in [22] and [23] proposed respectively a LES and a Direct Numerical Simulation (DNS) using the finite difference code "Incompact3d" with an impact distance of H/D = 2. The particularity of this latter work is the use of a conditional averaging post-processing procedure to compute the radial

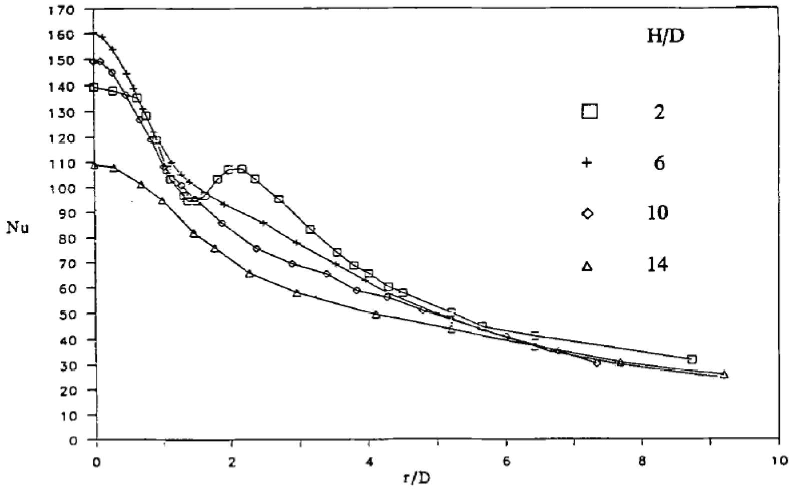


Figure 2.2 – Experimental radial Nusselt number distribution for different impact distances from Cooper et al. in [15]. Appearance of the secondary peak for the impact distance H/D = 2.

Nusselt number distribution. The most recent numerical results on the impingement jet cooling with an impact distance of H/D = 2 are those of Aillaud et al. in [19]. Thanks to the data sent by P. Aillaud from the Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS) and thanks to the easy access to available experimental data in the European Research Community on Flow, Turbulence and Combustion (ERCOFTAC) database, the specific configuration H/D = 2 and Re_{inj} = 23 000 has been chosen in this chapter.

In all the mentioned works, despite the variety of numerical methods, all the authors insist on the difficulty to solve the flow turbulent behavior of this configuration in order to capture the secondary peak in the radial Nusselt number distribution. In particular, they insist on the fact that a fully 3D turbulent and unsteady resolution of the problem is needed. It is therefore very challenging as we cannot use preliminary two dimensional (2D) computations to assess for the precision needed in our 3D simulations. In fact, despite the symmetry of revolution around the jet axis, the 2D numerical results will never be able to meet with the 3D experimental results because turbulence needs to be solved in the 3D space. This constrain imposes therefore: (i) to be able to handle a 3D geometry, (ii) efficient post-treatment procedures and above all (ii) important computational costs.

We begin this chapter in Section 2.2, with a preliminary study made during this PhD work using the commercial software ANSYS CFX and ANSYS Fluent. An analysis of the results obtained in this numerical framework is presented and we draw several conclusions that corroborate our findings from the literature. Then, we propose in Section 2.3, a numerical investigation realized using the finite element code Cimlib-CFD. To our knowledge, this is the first attempt to solve the impingement jet cooling problem using this finite element numerical framework. Indeed, we propose in this thesis a novel set of CFD tools to solve this complex aerothermal problem. To do so, we start in Section 2.3.1 with a description of the problem in terms of geometry and boundary conditions. Then, we bring into considerations in Section 2.3.2 the temporal and spatial discretizations chosen in this specific numerical framework. Furthermore, a mention of the experimental and numerical database on which we refer to compare our results is given in Section 2.3.3. In the next section (Section 2.3.4), we present the MATLAB program, implemented during this PhD work and used to post-treat the results on the radial Nusselt number distribution. Then, the results on velocity/pressure fields, vortex identification, velocity profiles and Nusselt number distribution are presented in Section 2.3.5 and compared to numerical and experimental references.

2.2 Preliminary study using the software ANSYS CFX and ANSYS Fluent

To begin this chapter on the single impingement jet cooling, we propose hereafter a preliminary study made during this thesis using the commercial software ANSYS CFX and ANSYS Fluent. This study allows testing a well known numerical framework in terms of meshing strategy, boundary conditions, solvers and post-treatment procedures, on the resolution of the impingement jet cooling. In particular, it makes it possible for us to confirm the conclusions from the literature [23–25] concerning the capture of the secondary peak in the Nusselt number distribution for the impact distance $\mathrm{H/D} = 2$, using these commercial software.

2.2.1 Context and problematic of the study

The goal of this study is to realize a preliminary study on the single impingement jet cooling using a numerical framework, currently used in aerospace research and industry. Indeed, ANSYS CFX and Fluent are well known software, often used to solve this type of aerothermal problem. Several numerical studies have already been made on the single impingement jet cooling using ANSYS software. To cite only one of them, the work of Wienand et al. in [25] shows interesting results in terms of radial Nusselt number distribution and velocity profiles in the boundary layer for different impact distances $\mathrm{H/D} = \{2, 6, 10, 14\}$.

The first idea here is to reproduce the study, for the configuration $\mathrm{H/D} = 2$, in order to validate our problem's setup in terms of geometry and boundary conditions. Secondly, this study allows us to asses for the practical aspects of computation in terms of (i) discretization, (ii) available numerical methods and (iii) possible post-treatment procedures. Finally, this study allows confirming the conclusions of the literature review about the origins of the secondary peak and, in particular, about the impossibility to capture it using steady-state RANS simulations.

2.2.2 Numerical methods

2.2.2.1 Structured 3D mesh

For both ANSYS CFX and ANSYS Fluent simulations, a structured hexahedral mesh is used. As seen previously in this chapter, the secondary peak in the Nusselt number distribution appears in the region between $r/\mathrm{D} \approx 1.5$ and $r/\mathrm{D} \approx 2.6$. Therefore, we build a mesh with a conic shape in order to capture more precisely the evolution of the turbulence in the free jet region. This mesh is presented in Figure 2.3. The height of the conic shape is $\mathrm{H}_c = 4.8\mathrm{D}$ and the radius of the conic shape is $\mathrm{R}_c = 3\mathrm{D}$.

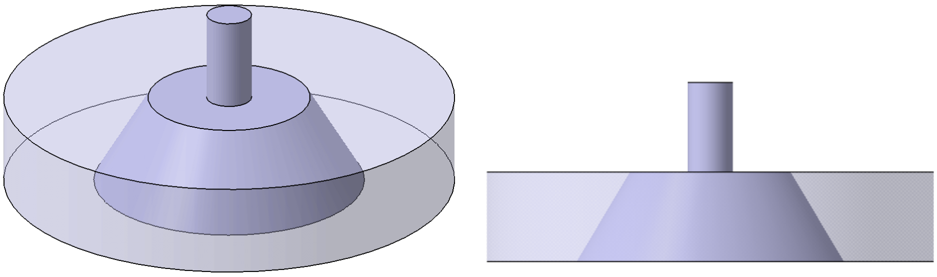


Figure 2.3 – Initial meshing strategy for ANSYS simulations.

For the mesh in the near plate area, in order to capture accurately the variation of temperature in the Z direction, the mesh size is chosen referring to the simulation of Wienand et al. in [25]. The height of the first layer on the plate is set to $z_1 = 13e^{-8}\ m$ and the height ratio of the adjacent mesh in two layers is set to 1.2. The successive layers are presented in Figure 2.4. With these parameters, we get a value of $z^+ = 0.74$.

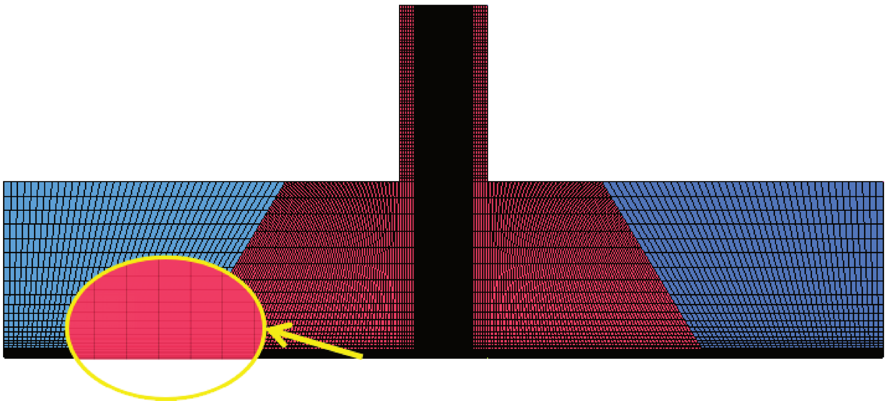


Figure 2.4 – Slice view of the hexahedral mesh used in ANSYS.

2.2.2.2 RANS approach for Navier-Stokes equations

In the ANSYS CFX and Fluent simulations, the discretization scheme is a finite volume scheme. The fluid motion solver is based on the RANS equations for the velocity-pressure fields. The flow fluctuating quantities are averaged in time using the Reynolds decomposition. In the Reynolds decomposition, the velocity $\mathbf{v}$ and the pressure p are separated into (i) a time-averaged component and (ii) a fluctuating component. Then, the RANS equations give us approximate time-averaged solutions of the Navier-Stokes equations. Therefore, the solution is indeed considered as a steady-state solution.

To balance the Navier-Stokes equations, an apparent stress term that takes into account the fluctuating velocity field is added to the equation. This term is generally referred as the Reynolds stress term. To approximate it, additional modeling is required and it has led to the creation of many different turbulence models in the past. One of them, the $k-\omega$ Shear Stress Transport (SST) turbulence model, was advised by the literature in our case and thus, it is the one we selected for this study. However, one drawback of this numerical method is that it has to be used with a prior knowledge of the flow turbulence properties.

2.2.2.3 The $k-\omega$ SST turbulence model

The $k-\omega$ SST model is a two-equations turbulence model based on the BaSeLine (BSL) $k-\omega$ model. It adds the transport of the turbulence shear stress in the definition of the turbulent viscosity to improve the prediction of the eddy-viscosity ν_t.

As explained in [26], in the BSL $k-\omega$ model, a blending function is used to combine (i) the Wilcox $k-\omega$ model introduced in [27] and (ii) the transformed $k-\epsilon$ model introduced in [28]. Doing so, it allows us (i) to retain the robust and accurate formulation of the Wilcox $k-\omega$ model in the near wall region but also (ii) to take advantages of the free-stream independence of the $k-\epsilon$ model in the outer part of the boundary layer. However, the BSL $k-\omega$ model still fails to properly predict the onset and the amount of flow separation from the smooth surfaces. As explained in [26], the Wilcox $k-\omega$ model and the transformed $k-\epsilon$ model lead to an over-prediction of the eddy-viscosity ν_t.

In the $k-\omega$ SST turbulence model however, the concept of full Reynolds-stress models is used. The shear-stress τ in the boundary layer is proportional to the turbulent kinetic energy k and defined as:

$$\tau = \rho_{fluid} a_1 k \tag{2.1}$$

where a_1 is a constant. The eddy viscosity is then defined as:

$$\nu_t = \frac{a_1 k}{max(a_1, \omega, SF_2)} \tag{2.2}$$

where F_2 is a second blending function defined as:

$$F_2 = max\left(\frac{2\sqrt{k}}{\beta' \omega z}, \frac{500\nu_{fluid}}{z^2\omega}\right) \tag{2.3}$$

and S an invariant measure of the strain rate. Since the eddy-viscosity value goes to infinity when the term SF_2 goes to zero, a limiter has been used to obtain a selection of eddy-viscosity and to let the eddy-viscosity function turn to original form $\nu_t = \frac{k}{\omega}$ in the boundary layer region.

2.2.2.4 Post-treatment procedure

Since a structured mesh is used in ANSYS, the value of each parameter is located on the mesh nodes. Therefore, in the calculation of the Nusselt number, the distance of the two values in the derivative $\left(\frac{\partial\theta_T}{\partial z}\right)_{z=0}$ is computed thanks to the height of the near-wall mesh. As shown in Figure 2.5, the average temperature T_0 on the plate is collected at the location called "line0", and the average temperature T_1 is collected at the location called "line1". The distance between these two lines is $h = z_1 = 13e^{-8}\ m$. The derivative of the normalized temperature θ_T is thus calculated as:

$$\left(\frac{\partial\theta_T}{\partial z}\right)_{z=0} = \frac{\theta_1 - \theta_0}{h} = \frac{T_1 - T_0}{(T_{plate} - T_{inlet})h} \tag{2.4}$$

Furthermore, since we use a cylindrical structured mesh, it is easy to obtain the azimuthal averaged values of the temperature. Indeed, the temperature at each radius location is only computed using the mesh points at this radius location.

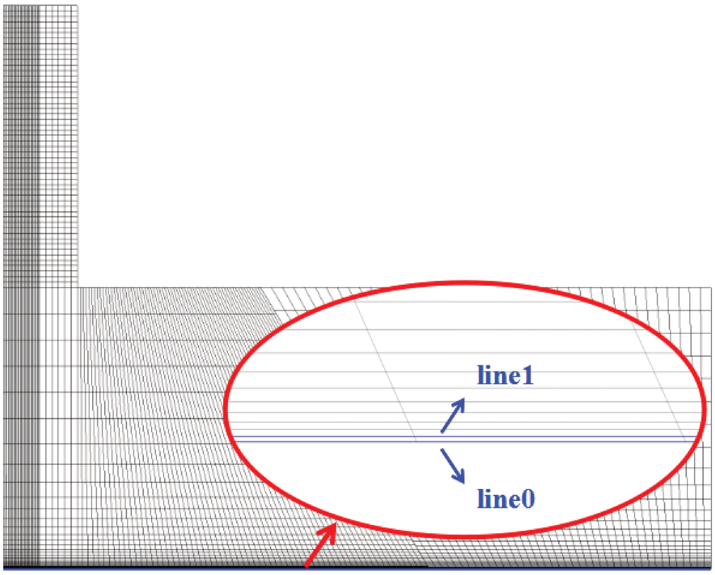


Figure 2.5 – Location of temperature extraction in the near wall region.

2.2.3 Results on the study

The numerical results presented here are the results obtained by Y. Yang during his Bachelor's thesis in [29]. They come from ANSYS CFX and ANSYS Fluent simulations that were run during his bachelor's thesis using the previously described numerical methods and post-treatment procedures. These numerical results are compared to the experimental results from Cooper et al. in [15].

2.2.3.1 Velocity and pressure field

The resulting velocity and pressure fields computed with ANSYS CFX are given in Figure 2.6. In this figure, we observe a time averaged solution of the problem. In fact, if we compare these results to the visualization from Cornaro et al in Figure 2.1, we observe that the RANS equations have a smoothing effect on the solution. Even if, by using the $k-\omega$ SST turbulence model, we indeed take into account the effect of turbulence in the resolution of the velocity and pressure fields, we cannot observe any influence of the turbulence on the macro-scale solution. In particular, as mentioned in the introduction of the present chapter, we cannot observe the primary nor secondary vortices, illustrated in Figure 2.14. The successive rebounds of the primary structures are not visible in these results.

2.2.3.2 Comparison on the velocity profiles in the boundary layer

Now that we have analyzed the velocity field macro-scale results, let's have a closer look on the velocity profiles in the boundary layer. The results on the velocity profiles at location $r/\mathrm{D} = \{0,1,2,3\}$ are presented in Figure 2.7. In this figure, we observe that, at location $r/\mathrm{D} = \{1,2,3\}$, the velocity profiles predicted by CFX are in good agreement with the experimental data. Then, for $r/\mathrm{D} = 3$, the results from CFX show an over-prediction of 8%. On the other hand, the results predicted by Fluent are accurate at point $r/\mathrm{D} = 1$ but shows a distinct over-prediction from the location between $r/\mathrm{D} = 2$ and $r/\mathrm{D} = 3$. The maximal gap between the radial velocity predicted by Fluent and the radial velocity from the experiment is observed at point $r/\mathrm{D} = 2$. At this point, the error is estimated at about 40%.

2.2.3.3 Comparison on the radial Nusselt number distribution

The results on the radial Nusselt number distribution computed by ANSYS CFX and ANSYS Fluent are presented in Figure 2.8. In general, both Fluent and CFX predict the tendency of the Nusselt number decrease with the radius. The results are in relative good agreement with the experimental data. Indeed, in the region $0 < r/\mathrm{D} < 1$, the predictions are both relatively accurate. The results calculated by Fluent fit well with the experimental data, and the results calculated by CFX are 9 % higher than the experimental results. However, in the region $1 < r/\mathrm{D} < 2$, the secondary peak in the Nusselt number distribution appears in the experimental data. On one hand, ANSYS-CFX predicts a fluctuation of the Nusselt number distribution in this region but this fluctuation cannot be identified as a proper secondary peak. On the other hand, ANSYS Fluent does not predict any fluctuation in the Nusselt number distribution in this area. The missing secondary peak in these simulations causes the inaccurate predictions in the next region $2 < r/\mathrm{D} < 3$. Finally, in the region $r/\mathrm{D} > 3$, when the radius increases, the effect of the missing secondary Nusselt peak decreases. The slope of the Nusselt number in both simulations becomes indeed comparable to the Nusselt number slope in the experimental data.

It is interesting to note that the location $r/\mathrm{D} = 2$ is, indeed, the location where the maximal error in the velocity profiles was found. Furthermore, it is also the location where the secondary peak appears in the Nusselt number distribution. Therefore, we can clearly see that the missing secondary peak in the radial Nusselt distribution is due to a poor resolution of the radial velocity in the boundary layer. As expected, this result illustrates the need to have a highly precise flow motion solution in the boundary layer in order to get the right convective heat transfer on the plate.

2.2.4 Conclusions and discussions on the study

To conclude, this study allowed us first, by running simulations on trustworthy commercial software, to test a computational framework currently used in aerospace research and industry. Furthermore, it made it possible for us to validate the problem's setup in terms of geometry and

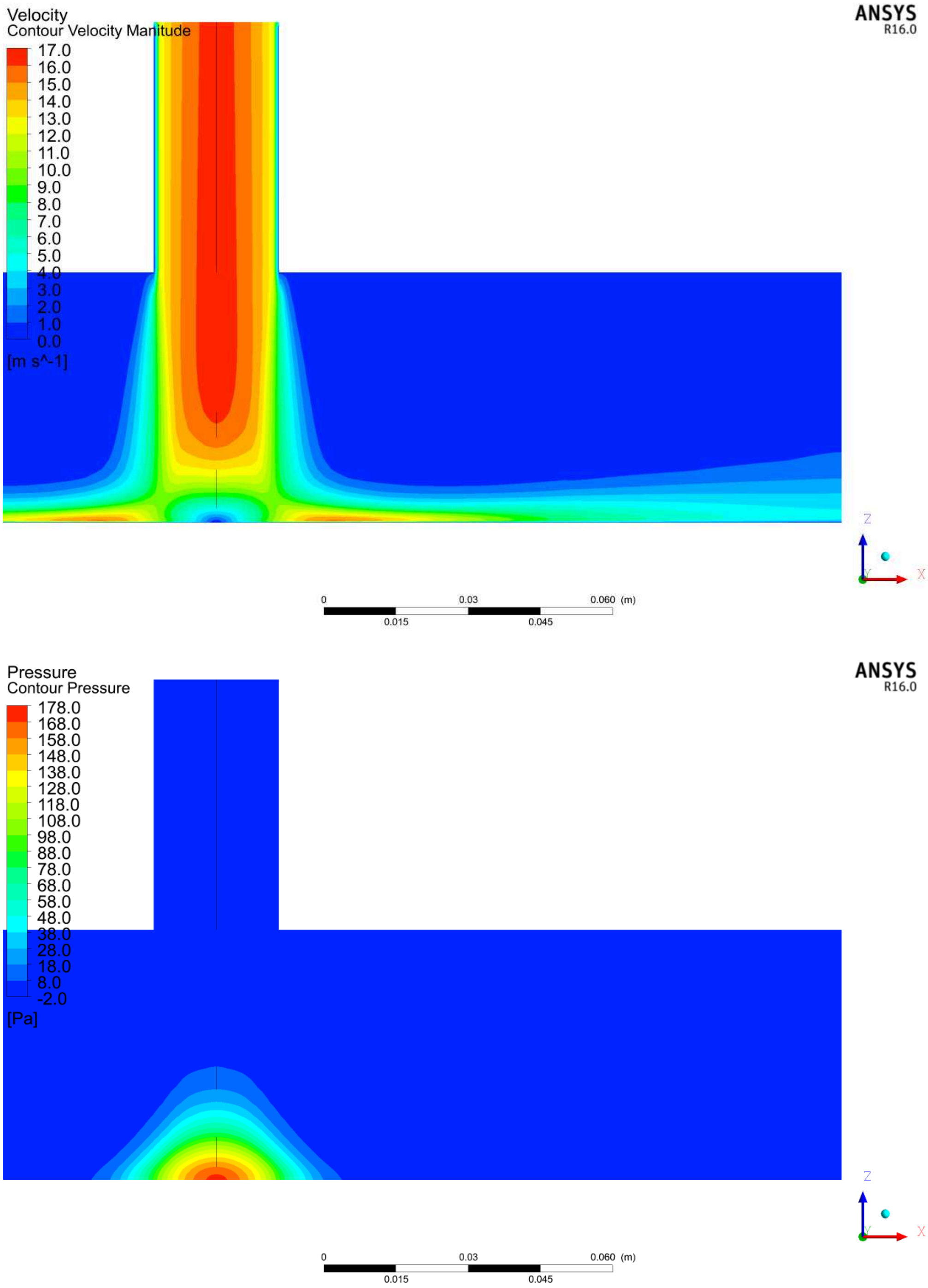


Figure 2.6 – Resulting velocity magnitude and pressure field for RANS $k-\omega$ SST simulations with ANSYS

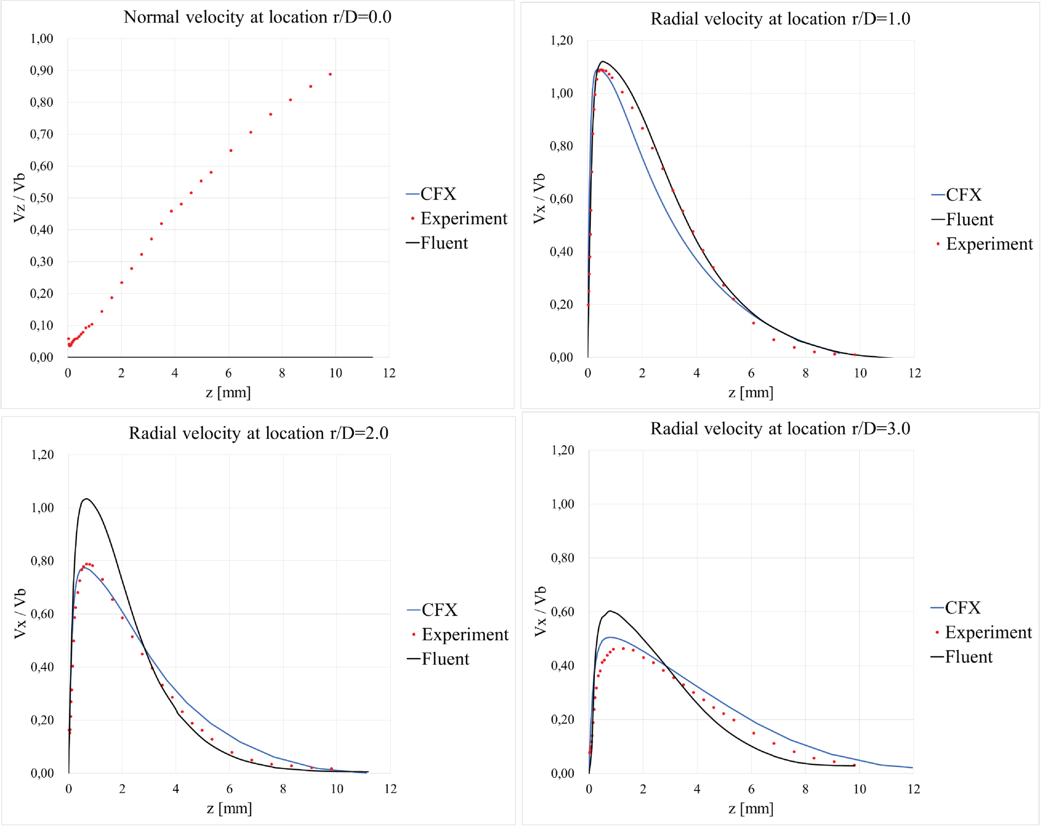


Figure 2.7 – Normal and radial and velocity profiles: results from Fluent (black color), from CFX (blue color) and experimental data (red color) from Cooper et al. in [15]

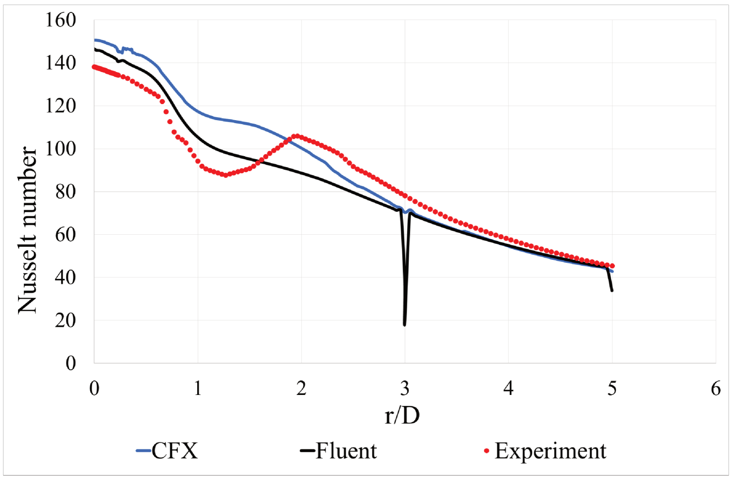


Figure 2.8 – Radial Nusselt number distribution: results from Fluent (black color), from CFX (blue color) and experimental data (red color) from Cooper et al. in [15]

boundary conditions. A structured type mesh was used here along with well known numerical methods like the time averaged RANS equations combined with the $k-\omega$ SST turbulence model. Finally, a well adapted post-treatment procedure designed to compute the radial Nusselt number distribution was proposed and validated in this computational framework.

This study also allowed us to draw several conclusions on the impingement jet cooling itself. In fact, from Section 2.2.3.1, we can conclude that it is impossible to obtain the primary nor the secondary vortices in the flow using the RANS equations. The smoothing effect cancels the rebound of the primary vortices on the plate. The consequence of these missing vortices was directly observable on the heat transfer results. Indeed, the resulting radial Nusselt number distributions from ANSYS CFX and Fluent were in relative good agreement with the experimental results except at the location of the secondary peak. A closer look on the velocity profiles inside the boundary layer showed the exact same results. Indeed, the RANS simulations allowed us to get good results on the velocity profiles in the boundary layer, except where secondary vortices should appear.

Finally, this preliminary study demonstrates the impossibility to get a precise result on the radial Nusselt number distribution of the impingement jet cooling using the ANSYS numerical frameworks. In particular, it is clearly impossible to capture the flow motion that will give birth to the secondary peak in the Nusselt number distribution. As mention in the introduction of the present chapter, in order to solve this aerothermal case, we need a fully 3D turbulent and unsteady resolution of the problem. The second part of this chapter aims at giving our developments and results in this direction using Cimlib-CFD.

2.3 Numerical investigations using the finite element code Cimlib-CFD

In this section, we propose to test another numerical framework, that breakout from the one presented in the previous section. This framework uses the finite element code Cimlib-CFD, developed by the Computing and FLuid (CFL) research group at the Centre de Mise En Forme des matériaux (CEMEF) MINES-Paristech laboratory. To our knowledge, this is the first attempt to solve the impingement jet cooling problem using this finite element numerical framework. This section gives the main outlines of our numerical investigations on this complex aerothermal case.

2.3.1 Geometry and boundary conditions

2.3.1.1 A fully 3D configuration

To begin this section, the 3D geometry of the problem is presented in Figure 2.9. The present configuration has an axial symmetry of revolution around the Z axis going through the center of the injection hole. As shown in previous studies and in particular by Dairay et al. in [23], a fully 3D turbulent resolution is needed to solve this complex aerothermal problem. In fact, considering that the jet is fully developed at the injection hole exit, it has been proven that the jet implies the development of turbulent structures that will evolved in the three spatial directions X, Y and Z axis.

Referring to previous works on the subject [14, 15, 18, 19, 23, 25], the nozzle to plate distance H is taken as two times the jet diameter D. In addition, we add an inlet tube of length 2D at the injection hole. The radial size of the domain is taken as 10D. However, in order to reduce the size of the domain and thanks to the cylindrical symmetry of the latter, we split the computation in half, keeping only the X axis positive sector. To do so, we impose a symmetric boundary condition on the plane (Y,Z) at $x=0$. Several numerical experiments made during this work have shown that this splitting had no influence on the final results.

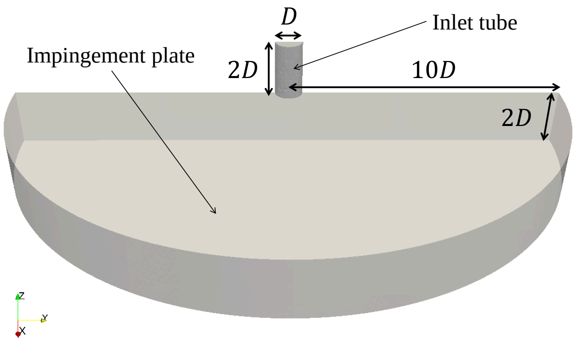


Figure 2.9 – Geometry of the computational domain

2.3.1.2 General boundary conditions

A representation of the problem boundary conditions is given in Figure 2.10. This figure represents a 2D slice of the 3D computation. The geometrical and physical parameters associated with these boundary conditions are given on Table 2.1; along with other physical parameters of the problem. The plate and the inlet tube edges are considered to be walls with no slip conditions. On the top boundary of the domain, a co-flow with a magnitude representing 5% of the bulk velocity is imposed. The choice of this value will be argue in the next section. At the outlet, to avoid any re-circulation of degenerated vortices, we choose to impose an output velocity for which the magnitude was determined after several tests on the geometry. At initialization, the velocity is null everywhere in the domain except on the boundary conditions.

As in previous experimental studies [14, 15], the jet temperature is taken at $\mathrm{T}_{inlet} = \mathrm{T}_{init} =$ 300 K. The plate is treated as an isothermal wall with a temperature $\mathrm{T}_{plate} = 330$ K. In this work and from a numerical point of view, we thus propose to solve a thermal convection-diffusion problem with only Dirichlet boundary conditions.

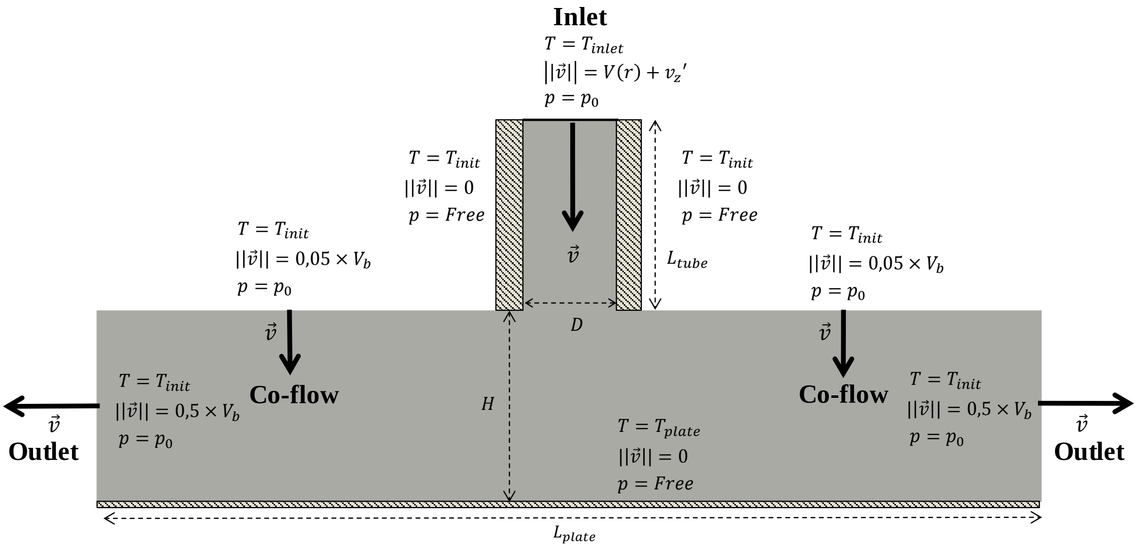


Figure 2.10 – Boundary conditions of the single impinging jet cooling

Parameter	Numerical value	Parameter	Numerical value
g	$9.81\ m.s^{-2}$	V_c	$17.5\ m/s$
β	$3.4e^{-3}\ K^{-1}$	V_b	$13.8\ m/s$
ν_{fluid}	$15.6e^{-6}\ m^2.s^{-1}$	D	$0.026\ m$
μ_{fluid}	$19e^{-6}\ Pa.s$	H	$0.052\ m$
ρ_{fluid}	$1.225\ kg.m^{-3}$	L_{plate}	$0.260\ m$
λ_{fluid}	$0.027\ W.m^{-1}.K^{-1}$	L_{tube}	$0.052\ m$
α_{fluid}	$20e^{-6}\ m^2.s^{-1}$	T_{inlet}	300 K
c_p	$1004\ J.kg^{-1}.K^{-1}$	T_{plate}	330 K
Re_{inj}	23000	p_0	$0\ Pa$

Table 2.1 – Geometrical and physical problem parameters

2.3.1.3 Co-flow boundary condition

For unconfined impinging jets, setting only a static pressure p_0 on the top free boundary as for standard outlets is usually not sufficient to obtain accurate simulations. In fact, re-circulation of the ambient air occurs on this top boundary leading to massive inflow and outflow. To counter this issue, Aillaud et al. in [19] suggest a specific treatment of this boundary condition by imposing a weak constant and perpendicular co-flow directed towards the plate. In [6], the same author also proposes a sensitivity analysis to demonstrate that: for a co-flow velocity magnitude inferior to 5% of the bulk velocity, the co-flow has no influence on: (i) the mean flow, (ii) the jet dynamic and (iii) the heat transfer. Furthermore, in the work of Hadziabdic et al. in [30], the authors also highlight that, within this co-flow velocity magnitude range, the jet flow region is not influenced by the co-flow. They explain this non-influence by the fact that the small intensity of the co-flow compared to the jet velocity allows the jet flow to act as a shield preventing any influence of the co-flow on the plate heat transfer. Furthermore, in [20], Grenson et al. propose to impose a velocity profile along the radial direction for this boundary condition. The mean value of the profile is around 15% of the bulk velocity but the profile itself presents the interest to be equal to zero close to the free jet region. However, it is equivalent to our configuration as we consider, in our geometry, a pipe thickness that cancel the co-flow velocity near the free jet region. Finally, following the argumentation of these three authors, and comparing our results with the numerical results of Aillaud et al in [19], the velocity magnitude of our co-flow is fixed at 5% of the bulk velocity V_b.

2.3.1.4 Turbulent inlet

In the present configuration, the inlet velocity boundary condition is of major importance because the jet needs to be fully developed at the injection hole exit. To treat this boundary condition, we first propose to impose a mean velocity profile for which the bulk velocity is given by:

$$V_b = \frac{Re_{inj} \times \nu_{fluid}}{D} \tag{2.5}$$

with ν_{fluid} the air kinematic viscosity. Following the work of Cooper et al. in [15] and other authors on the subject in [19, 23], the axial velocity is expressed using a specific power law profile for turbulent pipe flows. Indeed, we need to choose a profile law that is applicable to fully developed turbulent jets exiting a long tube. In direction Z, we have:

$$V(r) = \left(1 - \frac{2r}{D}\right)^{1/7.23} \times V_c \tag{2.6}$$

where r represents the distance to jet center axis and V_c is the centerline velocity. According to Cooper et al., it is given by the following expression:

$$V_c = \frac{V_b}{0.811 + 0.038(log(Re) - 4)} \tag{2.7}$$

Furthermore, to simulate a fully developed turbulent jet at the injection hole exit and because we do not simulate the flow inside the long experimental tube, we add synthetic turbulent perturbations which are superimposed on the mean velocity profile. These perturbations are computed using the expression given by Gautier et al. in [31]. We have:

$$v_z' = f_{mod}(r) \sum_{m=1}^{N} A_m(t)\, cos(m\theta + \phi_m(t)) \tag{2.8}$$

where θ is the angle on the inlet plane, m is an azimuthal mode and N is the number of excited azimuthal modes. The amplitude and phase $(A_m(t), \phi_m(t))$ are generated randomly up to a cutoff frequency. Referring to Dairay et al. in [22, 23], the modulation function $f_{mod}(r)$ is adjusted in order to match roughly the experimental conditions. It is expressed as follows:

$$f_{mod}(r) = A(R - r)exp(-\sigma(R - r)) \tag{2.9}$$

with $A = 0.7\sigma exp(1)$ and $\sigma = 1/(0.02R)$.

In order to impose this turbulent inlet boundary condition, one can explore other solutions in the literature. For example, a common way to impose this turbulent inlet is by adding isotropic homogeneous turbulence based on the approach of Bechara et al. in [32]. Doing so, the turbulence is injected over the whole inflow section in order to feed the core region of the tube with resolved turbulent content. Then, comparable to the solution proposed here but in a compressible fluid context, Bogey et al. in [33] propose to generate vortex rings in the flow field in the close vicinity of the tube's wall. Doing so, it allows starting up the boundary layer inside the tube in order for the flow to rapidly evolve into a turbulent state. Finally, other works like in [25] propose to compute a preliminary simulation of the flow inside a long inlet tube. This allows computing the development of turbulence inside a long tube in order to provide the unsteady inlet boundary condition of a fully developed jet to the impinging jet cooling simulation. However, the drawback of this solution is the need to have a well resolved preliminary computation that can be computationally demanding.

2.3.2 Temporal and spatial discretizations

2.3.2.1 Temporal discretization

In Cimlib-CFD, we use implicit time integration schemes concerning the temporal discretization of PDEs. Doing so, we thus circumvent the Courant–Friedrichs–Lewy (CFL) condition necessary for the convergence of explicit time integration schemes. Therefore, we do not have any numerical restriction on the choice of the simulation time step. The only thing that matters in this choice are physical considerations.

Indeed, the time step of the simulation is chosen in order to capture the unsteady flow behavior, in particular, at the injection hole exit where the generation of the main turbulent structures occurs. The goal here is to compute a mean value of the heat transfer by still capturing the unsteady flow characteristics that will drive the convective heat transfer on the plate. Therefore, the time step Δ_t is chosen with respect to the impinging frequency of the main large scale turbulent structures. The impinging frequency f_0 of these structures is computed thanks to the Strouhal number $St_D = 0.79$, the bulk velocity V_b and the diameter D of the injection hole (see [19] for more details). We have:

$$f_0 = \frac{St_D V_b}{D} \tag{2.10}$$

Then, we can deduce a period for these main large scale structures which is equal to $T_0 = 0.0024\ s$. After several numerical experiments, we decided to compute around 100 increments for each of these periods. Therefore, we choose a time step equals to $\Delta t = 0.00002\ s$. For a time step of $\Delta t = 0.00002\ s$, the maximum CFL condition is evaluated around 0.3 which gives us confidence on our starting hypothesis. Finally, to be able to have a representative simulation of the problem going from the beginning of the transient regime to an established unsteady regime, as in previous works [19, 20, 23], we decide to take around 20 periods of the main large scale structures for statistical analysis. These considerations give us a total simulation duration of $t_{total} = 0.1\ s$ counting thus 5 000 increments.

2.3.2.2 Initial meshing strategy

The reference mesh for the simulation is presented in Figure 2.11. It is a fixed mesh, adapted thanks to the well known Level-Set method mentioned in [34, 35] for example. This mesh counts 10 052 861 P1 elements and it is composed by different specific parts. The first specific part is a geometric cone shape of axis Z going through the center of the injection hole. This geometric form allows, on one hand, keeping a sufficient number of elements inside the inlet tube (see Figure 2.11 *a*)). And on the other hand, it allows refining the mesh in the central area of the domain where most of the turbulent structures will appear. Doing so, we keep an accurate aerodynamic resolution in the main region of interest but we also save an important number of elements from the area located far from the central region.

The mesh is then adapted where the aerodynamic boundary layer is susceptible to appear i.e. on the impingement plate (see Figure 2.11 *c*)). Because we use unstructured meshes, the mesh sizes can be different from one direction to the other. For this reason, we plot the diameter of the elements in Figure 2.11 *a*) and the minimal diameter of the elements in Figure 2.11 *b*). In this figure, we observe that we reach mesh sizes of order $10^{-6}m$ on the plate. Furthermore, as we want to capture the thermal activity on the plate, it is therefore necessary to have an important number of elements inside this boundary layer. For this reason, in Figure 2.11 *d*), we plot the dimensionless distance z^+ that will be introduced in Section 3.4.2 to show that, in the near wall region, we keep a value of $z^+ \leq 1$. For comparison, a similar mesh was proposed in the work of Wienand et al. in [25] with, instead, 20 prism layers on the plate to achieve an acceptable value for the dimensionless distance z^+.

2.3.3 Available experimental and numerical data

In this section, we focus our interest on the plate heat transfer results and their validation with respect to experimental and numerical references. Several experimental databases on Nusselt number distribution are available in the literature. In particular, data from the work of Baughn et al. and Cooper et al. in [14] and [15] on the radial Nusselt number distributions are given in the ERCOFTAC database. As we saw in introduction, Cornaro et al. in [16] also proposed an experimental study to visualize the impingement jet flow at different Reynolds numbers but also with different impact distances and different relative curvatures for the impingement plate. More recently, Fenot et al. in [18] also proposed an experimental database for the impingement jet cooling in our specific configuration.

Concerning numerical references, the recent results from Aillaud et al. in [19] on the impingement jet configuration H/D = 2 and Re = 23 0000 are used in this paper. Indeed, we refer to the database the author sent us for comparison.

2.3.4 Post-treatment procedure for the Nusselt number

From an industrial point of view, the main objective of this work is to evaluate the cooling performance of the impingement jet. To do so, we compute the radial Nusselt number distribution on the impingement plate. In the particular case of the impingement jet cooling, the Nusselt number

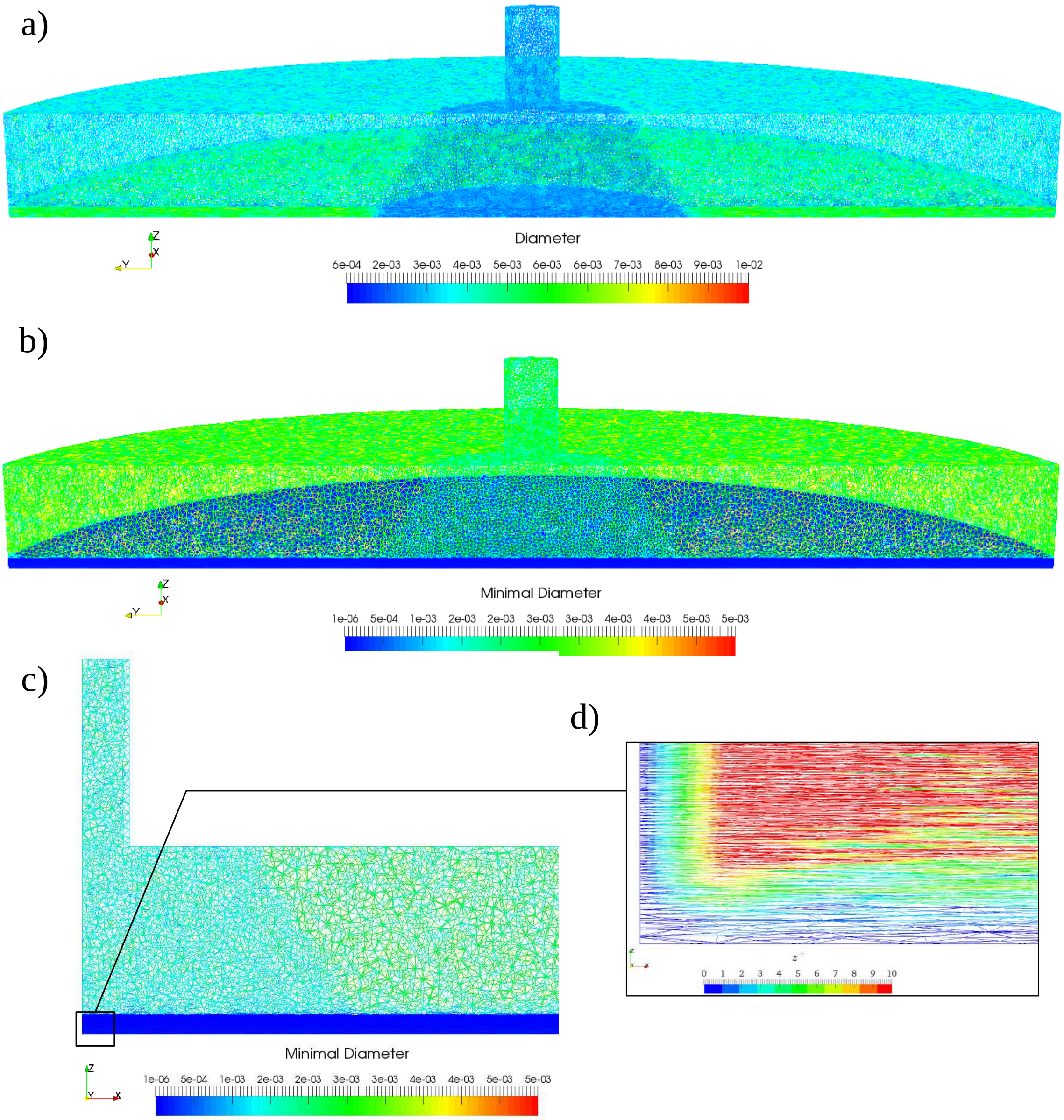

a)
Diameter
6e-04 2e-03 3e-03 4e-03 5e-03 6e-03 6e-03 7e-03 8e-03 9e-03 1e-02
b)
Minimal Diameter
1e-06 5e-04 1e-03 2e-03 2e-03 3e-03 3e-03 4e-03 4e-03 5e-03 5e-03
c)
d)
z^+
0 1 2 3 4 5 6 7 8 9 10
Minimal Diameter
1e-06 5e-04 1e-03 2e-03 2e-03 3e-03 3e-03 4e-03 4e-03 5e-03 5e-03
Z
Y
X

Figure 2.11 – Initial fixed adapted mesh. a) Element size distribution over the domain; b) Minimal element size distribution over the domain; c) Slice view of element diameters in plane (Z,X) at $y = 0$; d) Zoomed view of z^+ parameter on plane (Z,X) at $y = 0$

is defined by the following formula:

$$\mathrm{N}u = \left(\frac{\mathrm{L}_c h_{conv}}{\lambda_{total}}\right)_{z=0} = \frac{\mathrm{L}_c h_{conv}}{\lambda_{fluid}} \tag{2.11}$$

where L_c is the characteristic length of the problem; taken as the hole diameter D, h_{conv} is the convective heat transfer coefficient and λ_{total} is the fluid thermal conductivity. Since we use the Van Driest damping function in the computation of λ_{total}, its value on the plate at $z = 0$ is equals to λ_{fluid} (see Section 3.4.2 for more details).

2.3.4.1 Theoretical aspects

To evaluate the Nusselt number at each point of the plate, we start by normalizing the temperature. Thus, we work with the normalized temperature θ_T and we have:

$$\theta_\mathrm{T} = \frac{\mathrm{T} - \mathrm{T}_{inlet}}{\mathrm{T}_{plate} - \mathrm{T}_{inlet}} \tag{2.12}$$

Then, the heat flux from the hot surface to the fluid coolant is written as:

$$\phi_{conv} = h_{conv}\left(\mathrm{T}_{plate} - \mathrm{T}_{inlet}\right) = -\lambda_{fluid}\left[\left(\mathrm{T}_{plate} - \mathrm{T}_{inlet}\right)\left(\frac{\partial\theta_\mathrm{T}}{\partial z}\right)\right]_{z=0} \tag{2.13}$$

with

$$h_{conv} = -\lambda_{fluid}\left(\frac{\partial\theta_\mathrm{T}}{\partial z}\right)_{z=0} \tag{2.14}$$

Thus, we deduce the following formula for the Nusselt number:

$$\mathrm{N}u = \frac{\mathrm{L}_c h_{conv}}{\lambda_{fluid}} = -\mathrm{D}\left(\frac{\partial\theta_\mathrm{T}}{\partial z}\right)_{z=0} \tag{2.15}$$

With this expression, we note that, in this case, the local Nusselt number at each point of the plate is computed only by extracting the normalized temperature gradient at $z = 0$.

2.3.4.2 Azimuthal and temporal averaging procedures

In order to compare our results with experimental and numerical references, we need to compute an average value of the radial Nusselt number distribution. To do so, we first compute an azimuthal average value of the Nusselt number. To achieve this, we set 73 probes on the plate from which we extract the local Nusselt number $\mathrm{N}u$. These probes consists in 12 points spreading in 6 azimuthal directions which allow us to obtain 12 concentric circles around the plate center. The probes locations are presented in Figure 2.12. In this figure, the curved arrows represent the azimuthal averaging procedure. Following these arrows, an azimuthal average value is computed and a radial Nusselt distribution is drawn along the axis r/D in Section 2.3.5.4.

Then, since we solve an unsteady aerothermal problem, a temporal averaging procedure is also needed. Referring to [19, 20, 23] and as explained in Section 2.3.2.1, we compute the temporal average value of the Nusselt number over 20 cycles of the main large scale structure. The physical duration of this averaging procedure is therefore around $t_{mean} = 0.05\ s$. To illustrate the unsteady behavior of the simulation, we show in Figure 2.13 the local Nusselt number data extracted from the 12 sensors along direction 1.

Remark 1 *In Figure 2.13, the plot named "*$\mathrm{D}ir_0\ \mathrm{P}t_1$*" corresponds to the temporal data extracted from the central point of the plate.*

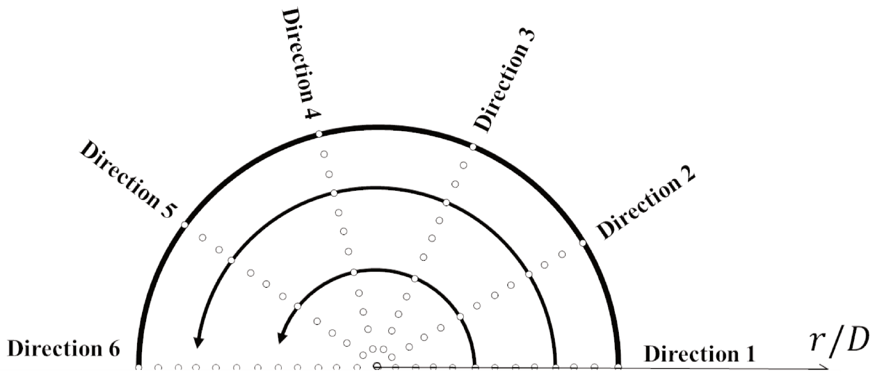


Figure 2.12 – Probe location on the plate

In Figure 2.13, we can observe a periodic behavior of the flow on the impingement plate. Furthermore, we observe that the main turbulent structures cycles are well captured by our post-treatment procedure. From the beginning of the established unsteady regime, we can count around 20 cycles of the main turbulent structures. Then, we also observe the propagation of the cooling in the radial direction over time. In fact, before increment 3 000, not all the plate has been affected by the cooling as we observe that some sensors do not detect any temperature gradients. As in experimental studies, the temporal averaging procedure needs to start when the permanent regime is well established and thus, after all the sensors have been affected by the cooling. We choose 3 000 as the starting increment for the averaging procedure.

The main advantage of this post-treatment procedure using probes is that it allows following in real time the results of the simulation. In fact, each sensor output and update its data at every increment in text files readable by another program. We can thus use MATLAB to read these data whenever we want to check for the results. Furthermore, this post-treatment procedure is independent of the mesh used and therefore, very practical to handle. However, the draw back is that we only take into account for the heat transfer occurring at the sensors positions. Therefore, the user has to ensure that enough data are collected on the plate in order to have a representative azimuthal average of the heat transfer.

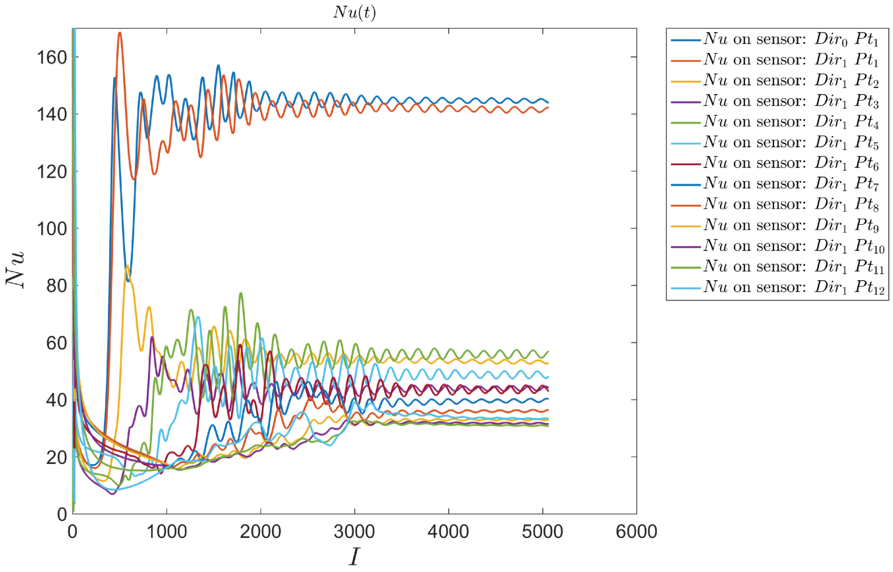


Figure 2.13 – Temporal statistics for Nu in Direction 1

2.3.5 Results and discussions

The results on the velocity and pressure fields $(\mathbf{v}, p)$ are obtained via the implicit solving of the unsteady incompressible Navier-Stokes equations. The incompressible assumption stands here because the maximum velocity encountered in the flow is around 20 m/s, giving a Mach Number around 0.06 which less than the 0.3 limit for incompressible flows. The solver uses the stabilized finite element formulation VMS and details about its resolution will be given in the next chapter. Concerning the temperature T, we solve the convection-diffusion equation using the SUPG numerical scheme to stabilized the standard Galerkin formulation. As well, details on the resolution of this finite element problem will be given in the next chapter.

In this section, we present the results obtained with Cimlib-CFD using the previously described boundary conditions, temporal/spatial discretizations and post-treatment procedures. First, to introduce the general flow behavior of this specific configuration, we present the obtained velocity/pressure fields over time. Then, to complete the flow analysis, a post-treatment procedure on vortex identification is proposed. Furthermore, the velocity profiles inside the plate boundary layer are analyzed and compared to experimental data. From the velocity profiles validation, we complete our aerothermal analysis by a comparison of the resulting radial Nusselt number distribution with experimental and numerical references.

2.3.5.1 Instantaneous velocity and pressure fields

The resulting velocity field over time is presented in Figure 2.14. As mention in the introduction of the present chapter, the fundamental phenomenon that occurs during the impingement jet cooling concerns the rebound of the primary structures when the jet impacts the hot plate. The first primordial rebound takes place during the transitional flow at increment I = 700. As shown in Figure 2.14, it generates secondary vortices in the primary vortex opposite direction near the wall and induces a re-injection of cold fluid on the plate. Furthermore, we can observe at increment I = 3 000 that the rebound phenomenon does not disappear over time. Indeed, it evolves into a periodic wave that is caused by the successive rebounds of the successive primary structures exiting the injection hole during the simulation.

The resulting pressure field over time is presented in Figure 2.15. We observe the same flow structure than the velocity field. In particular, between increment I = 600 and I = 700, we can follow the first rebound of the primary vortex on the plate. In this area, the inversion of the pressure sign is a reliable parameter to identify the vortex directions that will be highlighted in the next subsection. The birth of secondary vortices and thus the injection of cold fluid on the plate seems to clearly start at increment I = 600. Then, as expected, at increment I = 3 000, we recover the periodic wave caused by the successive rebounds of the primary structures.

2.3.5.2 Vortex identification

To complete this aerodynamic analysis, we propose here to use the vortex identification technique to theoretically identify the primary and the secondary vortices in the impinging jet. To do so, we use the well known Q-criterion method, introduced by Haller et al. in [36], where Q is defined by:

$$\mathrm{Q} = \frac{1}{2}\left[(tr(\nabla\mathbf{v})^2 - tr(\nabla\mathbf{v}\cdot\nabla\mathbf{v})\right] = \frac{1}{2}||\Omega_{\mathbf{v}}||^2 - ||\mathrm{S}_{\mathbf{v}}||^2 \qquad (2.16)$$

where $\Omega_{\mathbf{v}}$ is the skew-symmetric part of the velocity gradient tensor and $\mathrm{S}_{\mathbf{v}}$ is the symmetric part of the velocity gradient tensor. In Cartesian coordinates, by simplification of the previous expression, we have the following definition of Q:

$$\mathrm{Q} = \left(\frac{\partial v_y}{\partial y}\frac{\partial v_z}{\partial z} - \frac{\partial v_y}{\partial z}\frac{\partial v_z}{\partial y}\right) + \left(\frac{\partial v_x}{\partial x}\frac{\partial v_y}{\partial y} - \frac{\partial v_x}{\partial y}\frac{\partial v_y}{\partial x}\right) + \left(\frac{\partial v_z}{\partial z}\frac{\partial v_x}{\partial x} - \frac{\partial v_x}{\partial z}\frac{\partial v_z}{\partial x}\right) \qquad (2.17)$$

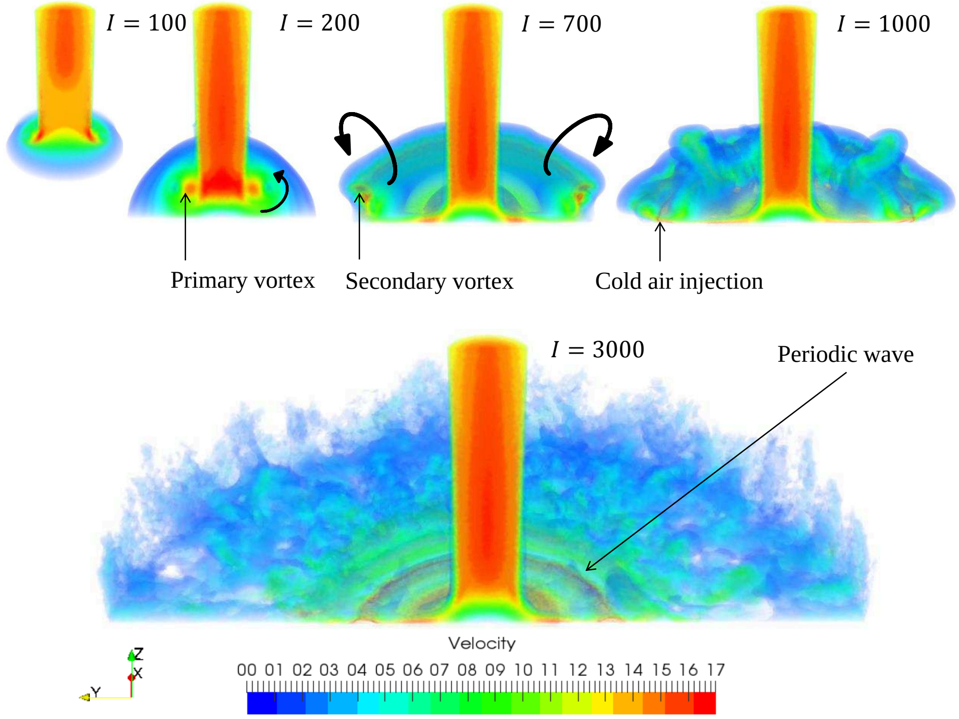


Figure 2.14 – Resulting velocity field over time

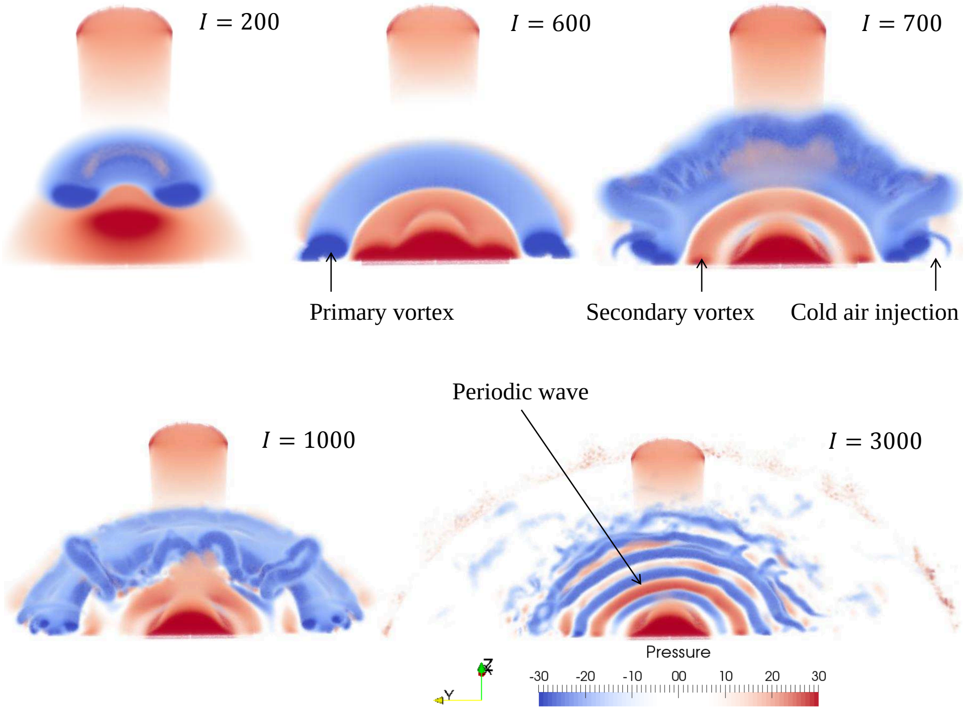


Figure 2.15 – Resulting pressure field over time

Referring to the work of Dairay et al. in [23], we extract the contour of iso-values of Q such that $Q = 100\frac{{V_b}^2}{D}$. The resulting contour is given in Figure 2.16 (without considering the colors).

As explained in the previous subsection, the primary vortices are identified by the flow vortices rotating clockwise around the Y axis. In the same way, the secondary vortex are identified by the flow vortices rotating anticlockwise around the Y axis. Therefore, to identify the vortices, we compute the vorticity $\boldsymbol{\omega}$ of the flow defined by:

$$\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z) = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}, \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}, \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right) \tag{2.18}$$

In particular, we extract the vorticity component around the Y axis called ω_y. The sign of ω_y allows us to know if the flow is rotating clockwise or anticlockwise around the Y axis. In fact, if $\omega_y > 0$, then the flow is rotating clockwise and if $\omega_y < 0$, then the flow is rotating anticlockwise. The result are presented in Figure 2.16. In this figure, the red color corresponds to $\omega_y > 0$ and thus, is associated to a primary vortex structure; as well, the blue color corresponds to $\omega_y < 0$, and thus, is associated to a secondary vortex structure.

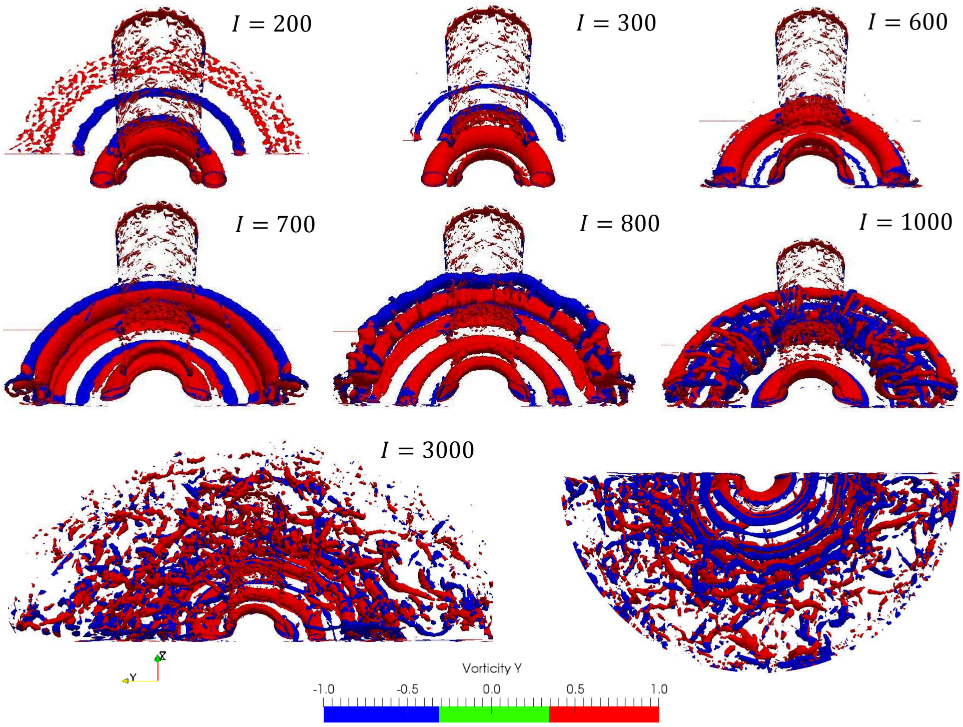


Figure 2.16 – Resulting instantaneous vortex identification over time. Red color corresponds to $\omega_y > 0$ (primary vortex). Blue color corresponds to $\omega_y < 0$ (secondary vortex).

In Figure 2.16, from increment I = 200 to increment I = 600, we observe the rebound of the first primary vortex on the impingement plate. Then, from increment I = 600 to increment I = 1000, we observe the birth of the secondary vortices on the plate (see the blue color at increment I = 600). The secondary vortices then evolve into something more chaotic until increment I = 3000. However, even if the structure of the fluid is less clear at increment I = 3000, if we look at the bottom of the plate (see last picture in Figure 2.16), we find the previously mentioned periodic wave that induces the injection of cold fluid air on the plate. Indeed, the circular blue regions on this picture can be considered as cold spots, i.e. places where the plate meets with cold fluid.

2.3.5.3 Velocity profiles in the boundary layer

Now that we have analyzed the overall flow behavior of the impingement jet, we propose here to take an interest in the flow velocity profiles in the boundary layer formed by the impact plate. Indeed, this region is of major importance because this is where the heat transfer between the plate and the fluid occur. We base our aerodynamic analysis on the experimental study of Cooper et al. in [15] and we refer to the experimental data from the ERCOFTAC database.

For the reader's understanding, Figure 2.17 shows a schematic representation of the velocity profile setup. To obtain them, we place 35 sensors distributed along the Z direction at each r/D position. The sensors' positions in the Z direction are chosen with respect to the experimental data which are available for a height going from $z/\mathrm{D} = 0$ to $z/\mathrm{D} = 0.4$ (i.e. $z = 0.0104m$). At each iteration, the radial and normal velocity components are extracted from the sensors. Then, a temporal averaging procedure is done to obtain the time averaged velocity profiles at each r/D position.

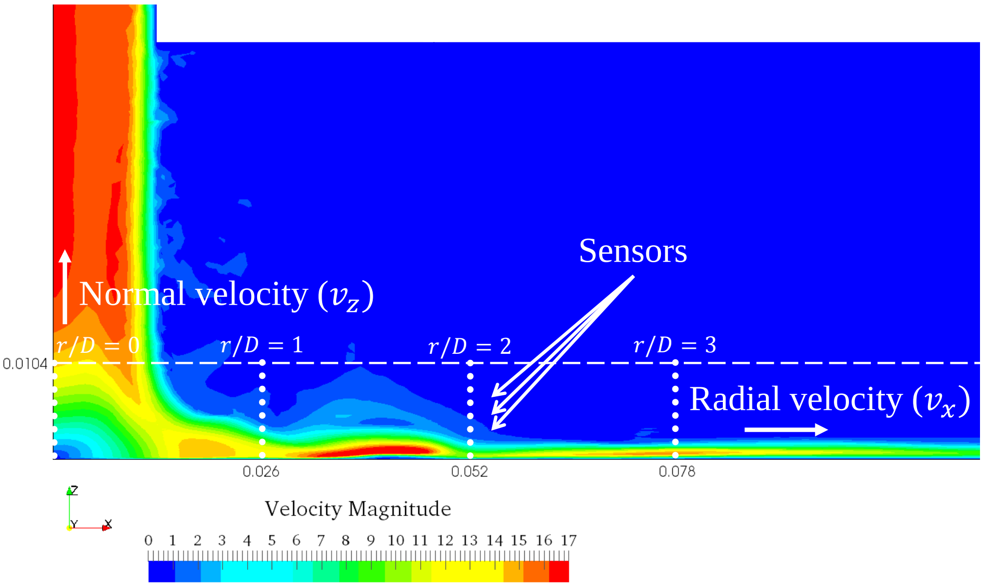


Figure 2.17 – Schematic representation of the extracted radial and normal velocity profiles

We first analyze the velocity profile in the boundary layer at $r/\mathrm{D} = 0$. At this position, the velocity component given by the experimental data is the normal velocity (i.e. the velocity component along the Z axis). The results are presented in Figure 2.18. In this figure, we observe that our numerical results are in good agreement with the experimental data. Indeed, the central point of the plate seems to be well resolved by our numerical tools. This observation gives us confidence on the fact that the inlet mean velocity profile is correct and well imposed at the tube's entry.

Then, the experimental study gives the velocity profiles of the flow at 3 other positions: $r/\mathrm{D} = 1$, 2, and 3 (i.e. $r = 0.026m$, $0.052m$, and $0.078m$). This time, the velocity component is the radial one and the results are given in Figure 2.19. In this figure, we compare the resulting velocity profiles to the experimental ones. At each r/D position, the numerical result respects the expected order of magnitude and corresponds to each experimental plot. In particular, in direction $r/\mathrm{D} = 1$ the results fit well with the experimental data.

However, the results tend to be less accurate when we go further from the plate center, i.e. when we look at direction $r/\mathrm{D} = 2$ and $r/\mathrm{D} = 3$. This loss of accuracy can be explained by the unclear definition of the output velocity on the side of the domain (see $\overrightarrow{v}$ on the Outlet in Figure

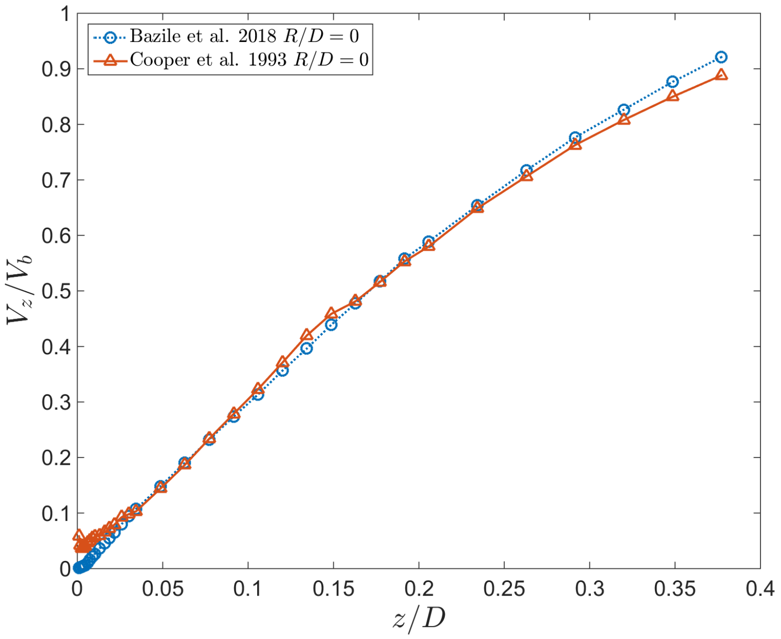


Figure 2.18 – Normal velocity profiles in the boundary layer at $r/\mathrm{D} = 0$

2.10). Indeed, the location of this outlet boundary condition remains unclear in the literature and the width of the domain can vary from a study to the other. During this work, a smaller geometry of radial size 5D was tested and the results got better when increasing the size of the domain to 10D. Doing so, we indeed moved apart the outlet boundary condition from the plate center and thus, reduced its effects on the flow. However, the effect of this boundary condition still seems to play an important role and this problem should be addressed as a perspective.

Furthermore, we have good agreement with experimental data in the near plate region. However, going away from the plate, the results tend to be less accurate. This shows that the mesh size in the boundary layer plays a crucial role on the velocity profiles results. We indeed used anisotropic elements to be able to maximize the resolution inside this boundary layer in the Z direction in order to capture the thermal gradients on the plate, without having prohibitive computational costs inherent to isotropic meshes. In our case, anisotropic mesh adaptation seems to be the best candidate to solve this problem.

2.3.5.4 Radial Nusselt number distribution

To finish the aerothermal analysis, we propose here to compare our results in terms of radial Nusselt number distribution with experimental and numerical references. It is first important to recall that, as shown in Section 2.3.5.1, Section 2.3.5.2 and in previous works [37], the generated periodic wave plays an important role in the convective heat transfer. In fact, the re-injection of cold air created by the secondary vortices enhances locally the cooling and thus generates a secondary peak in the radial Nusselt number distribution. This secondary peak is visible in Figure 2.20. It appears experimentally at $r/\mathrm{D} \approx 1.3$ and reaches its maximum at $r/\mathrm{D} \approx 2$. The results obtained thanks to the post-treatment procedure developed in Section 2.3.4 are given in Figure 2.20.

In this figure, we observe that the impact point is, again, well resolved by the proposed numerical framework. Looking at the result obtained at the impact point in the analysis of the velocity profiles, this result was expected. Indeed, in our case, the convective heat transfer are dominant.

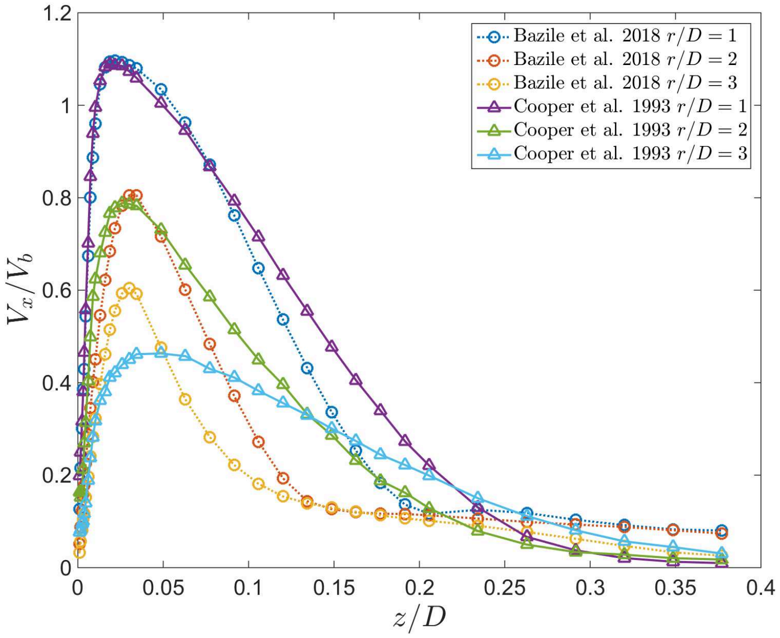


Figure 2.19 – Radial velocity profiles in the boundary layer at $r/\mathrm{D} = 1, 2, \& 3$

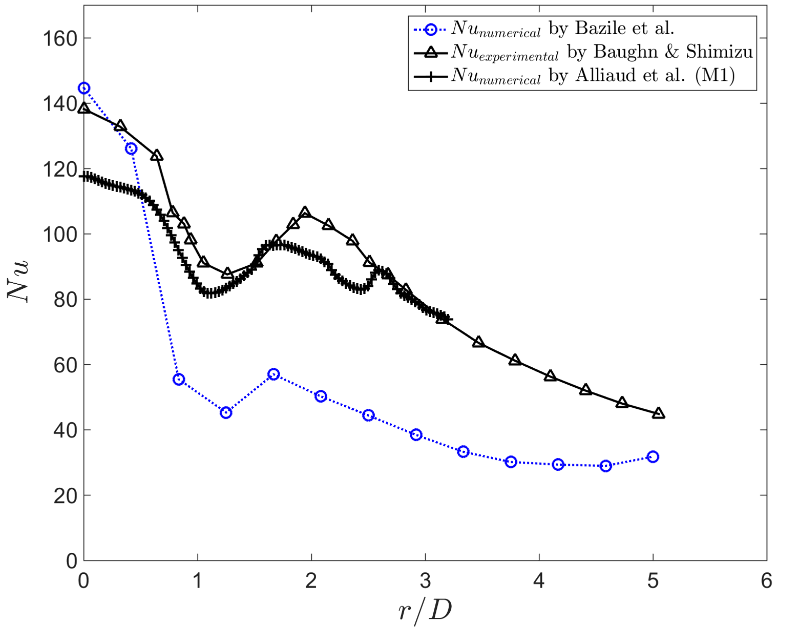


Figure 2.20 – Comparison of radial Nusselt number distribution with an experimental reference from Baughn et al. in [14] and with a numerical reference from Aillaud et al. in [19].

Therefore, the results on the velocity profiles in the boundary layer majorly drive the heat transfer results on the plate. Consequently, the observations made on the velocity profiles still hold for the analysis of the Nusselt number distribution. A loss of accuracy is observed when we get away from the plate center and this can be explained by both (i) the outlet free boundary condition and (ii) the mesh topology in the boundary layer.

Furthermore, the fact that our prediction of the Nusselt number is below the expected result can also be explained by the lack of mixing in the boundary layer. Indeed, the more the flow will be mixed in the boundary layer, the higher the convective heat transfer will be and thus, the higher the Nusselt number will get. According to the literature on this subject, this mixing is majorly driven by the inlet turbulent model used at the tube's entry. As a perspective, it would also be interesting to test other ways to impose the turbulent inlet boundary condition and observe their effects on the Nusselt number distribution.

Despite that, in Figure 2.20, we observe that our results allow in a certain way to capture the secondary peak in the radial Nusselt number distribution. In fact, even if we are quite far from the expected radial distribution, we observe a secondary peak at the same position that the experimental results. This shows that our unsteady simulation is able, in a certain way, to capture the complex heat transfer on the plate. This was in fact expected from the results in Section 2.3.5.2 where we clearly showed our ability to capture the complex turbulent structures in the overall analysis of the flow behavior.

2.4 Conclusion

To conclude, this chapter made it possible for us to better understand the complexity of the physical phenomenon occurring during the impingement jet cooling. We proposed in this chapter two different numerical frameworks to solve this complex aerothermal problem. The first one used ANSYS CFX and ANSYS Fluent and the second one used Cimlib-CFD. The preliminary study using the ANSYS framework clearly showed the need to have a fully 3D turbulent and unsteady resolution of the problem in order to capture the very specific heat transfer distribution of the impingement jet cooling on the impact plate. To do so, we then proposed numerical investigations around this complex aerothermal case using the finite element code Cimlib-CFD.

In Cimlib-CFD, we first proposed a description of the problem in terms of geometry and boundary conditions. In particular, an argumentation on the top co-flow boundary condition and a practical way to impose a turbulent inlet at the tube's entry were developed. Then, we explained the choices of spatial and temporal discretizations for this aerothermal simulation and a post-treatment procedure on the Nusselt number was developed.

The overall observations on the results obtained using Cimlib-CFD showed our ability to capture the global turbulent behavior of the flow. Indeed, the instantaneous velocity and pressure fields clearly showed the rebound of the primary vortices on the plate creating secondary vortices in the opposite direction. In particular, these vortices were identified using a vortex identification method that corroborated our results. With this phenomenological analysis, we managed to explain the re-injection of cold fluid air on the plate and thus, the appearance of the secondary peak in the radial Nusselt number distribution.

From the analysis made on the velocity profiles in the boundary layer but also, from the analysis made on the Nusselt number distribution on the plate, the results showed that the impact point was well resolved by the proposed numerical framework. However, from these results, several perspectives can be proposed for future works. In fact, the influence of the outlet boundary condition still seems to play a crucial role on the flow motion in the boundary layer. The location of this boundary condition and the way to impose it in the finite element framework is still an open question. Furthermore, the mesh in the boundary layer can still be improve in order to capture more precisely both (i) the velocity profiles in the boundary layer and (ii) the thermal gradient on the plate. To tackle this problem, anisotropic mesh adaptation seems to be the best solution.

Finally, even if we are confident that the proposed synthetic perturbations model for the turbulent inlet boundary condition is, indeed, well suited for our simulation, other solutions can be explored. Among others, we propose for future works a preliminary computation of the flow inside a long inlet tube in order to provide the turbulent inlet boundary condition of a fully developed jet. It will then be interesting to analyze the effects of this inlet boundary condition on the Nusselt number distribution.

In the next chapter, we will present the set of numerical tools used to solve this kind of highly convective aerothermal problem in Cimlib-CFD. In particular, we will propose additional numerical models to reinforce the aerothermal coupling of the simulation. These numerical tools will be validated on the benchmark case of the 3D ventilated cavity and different aerothermal couplings will be tested in order to choose the best one for our simulations.

Résumé du chapitre en français

Ce chapitre est consacré à l'étude aérothermique d'un jet non-confiné, isotherme, impactant verticalement sur une plaque à refroidir. Ces dernières années, de nombreuses études numériques et expérimentales ont été réalisées sur ce cas d'étude. En particulier, pour une distance d'impact H/D = 2 et un nombre de Reynolds d'injection supérieur à 10 000, il a été remarqué que la turbulence de l'écoulement générait une carte thermique très spécifique sur la plaque. En effet, l'apparition de tourbillons secondaires dans l'écoulement va favoriser localement les échanges thermiques convectifs et créer un second pique dans la distribution radiale du nombre de Nusselt. Dans tous les travaux mentionnés, malgré la grande variété des méthodes numériques utilisées, les auteurs insistent sur la difficulté de résoudre cette configuration aérothermique afin d'obtenir une distribution radiale du nombre de Nusselt en corrélation avec les études expérimentales. En particulier, ils mentionnent l'importance de réaliser une simulation turbulente et instationnaire en 3D qui demande (i) de gérer des géométries complexes en 3D, (ii) d'avoir des méthodes de post-traitement efficaces et (iii) d'avoir la possibilité d'utiliser d'importantes ressources de calcul.

Dans ce travail, nous proposons d'utiliser un nouvel ensemble d'outils CFD pour résoudre ce problème aérothermique. C'est, à notre connaissance, la première fois que ce cas est réalisé avec les outils numériques que nous développons au laboratoire. Pour cela, nous proposons d'abord une description du problème et de ses conditions aux limites. Ensuite, le choix des discrétisations spatiales et temporelles est développé dans notre contexte numérique. Par la suite, nous proposons une méthode de post-traitement développée sur MATLAB et implémentée pendant la thèse qui permet d'obtenir en temps-réel la distribution radiale du nombre de Nusselt sur la plaque.

Les résultats sur le comportement global de l'écoulement nous permettent de mettre en évidence la capacité de nos méthodes numériques à capturer le comportement turbulent de l'écoulement. En particulier, les visualisations instantanées des champs de vitesse et de pression permettent d'observer le rebond des structures primaires sur la plaque, donnant naissance aux tourbillons secondaires. De plus, le développement d'une méthode d'identification de tourbillons nous permet de corroborer ces résultats. Avec cette analyse, nous montrons notre capacité à expliquer l'apparition du second pique dans la distribution radiale du nombre de Nusselt.

À partir de l'analyse faite sur les profiles de vitesse dans la couche limite mais également, à partir de l'analyse faite sur la distribution du nombre de Nusselt sur la plaque, les résultats montrent que le point d'impact du jet est bien résolu avec les outils numériques proposés. Cependant, venant de ces résultats, certaines perspectives peuvent être proposées. En effet, l'influence de la condition aux limites de sortie semble toujours jouer un rôle crucial sur le déplacement du fluide dans la couche limite. La localisation de cette condition aux limite et la manière de l'imposer dans un framework éléments finis est toujours une question ouverte à ce jour. De plus, le maillage peut également être amélioré dans la couche limite dans le but de capturer plus précisément (i) les profiles de vitesse et (ii) les gradients thermiques sur la plaque. Pour résoudre cela, l'adaptation anisotrope de maillage semble être la meilleure solution.

Enfin, même si nous sommes confiant que la proposition apportée pour générer des perturbations synthétiques turbulente à l'entrée du jet est, en effet, la mieux adaptée à notre contexte numérique, d'autres solutions peuvent être explorées. Parmi d'autres, nous proposons dans le futur d'effectuer un calcul préliminaire d'un jet dans un long tube d'entrée de manière à fournir un jet turbulent développé en entrée de la simulation du refroidissement par impact. Il sera alors intéressant de regarder comment la distribution du nombre de Nusselt est affectée par cette nouvelle condition aux limites.

Chapter 3

FEM for Turbulent Flows with Heat Transfer

Contents

3.1 Introduction

In this chapter, we introduce the numerical methods used to solve aerothermal problems for turbulent flows in Cimlib-CFD. It is first important to mention here that we solve decoupled aerothermal problems that require to solve two different PDE systems. The first PDE system is dedicated to the implicit solving of the unsteady incompressible Navier-Stokes equations to compute the velocity and pressure fields $\mathbf{v}$ and p. This solver uses the stabilized finite element method introduced in [13, 38–40] and applied by Hachem et al. in [41]. In particular, the VMS formulation used in the latter paper shows to be favorable for simulating flows in the Reynolds number range of the present study. Indeed, the assumption of a convection dominant problem still hold in our case since we deal with turbulent flows associated to high Reynolds numbers. The solving of this first PDE system is the subject of Section 3.2.

The second PDE system is dedicated to the solving of the convection-diffusion equation for the temperature u. It takes, as an input and from the previous resolution, the resulting velocity $\mathbf{v}$ in the convection term. Considering the thermal diffusivity a_{fluid} of the problem, we compute it from the physical parameters given in Table 2.1. We get:

$$a_{fluid} = \frac{\lambda_{fluid}}{\rho_{fluid}\, c_p} = 22e^{-6}\ m^2.s^{-1} \tag{3.1}$$

Therefore, with a thermal diffusivity of this magnitude, the convective heat transfers are considered to be dominants during the cooling. We thus refer to the use of the well known SUPG scheme introduced in [13] to stabilize the standard Galerkin formulation. The resolution of this thermal finite element problem is the subject of Section 3.3. It is written in a generalized way using u to refer to the solution of the problem that is, in our case, the temperature.

Furthermore, as we solve decoupled aerothermal problems, additional numerical models are needed in order to achieve aerothermal coupling in the simulation. In this work, we use two different numerical models simultaneously: (i) The Sutherland law and (ii) the Smagorinsky Model (SM). These additional numerical models are presented in Section 3.4.

Finally, in Section 3.5, we propose a numerical study of the 3D ventilated cavity in order to validate the numerical methods implemented in Cimlib-CFD and the additional numerical models that we add in order to achieve aerothermal coupling. This study is based on a well documented benchmark and allows, in particular, to test different aerothermal coupling configurations in order to choose the best one for our study.

3.2 The incompressible Navier-Stokes equations

3.2.1 Governing equations

To fix the notations, let $\Omega \subset \mathbb{R}^d$ be the fluid domain, where d is the space dimension, and $\partial\Omega$ its boundary. The strong form of the incompressible Navier Stokes equations reads:

$$\begin{cases} \rho\,(\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{f} \\ \nabla\cdot\mathbf{v} = 0 \end{cases} \tag{3.2}$$

where ∇ is the gradient operator, $t \in [0, t_{total}]$ is the time, $\mathbf{v}(\mathbf{x}, t)$ is the velocity, $p(\mathbf{x}, t)$ is the pressure, $\rho = \rho_{fluid}$ is the density and $\mathbf{x}$ is the position vector. The Cauchy stress tensor $\boldsymbol{\sigma}$ for a Newtonian fluid is given by:

$$\boldsymbol{\sigma} = 2\mu\,\boldsymbol{\varepsilon}(\mathbf{v}) - p\,\mathbf{I}_d, \tag{3.3}$$

with $\mathbf{I}_d$ the d-dimensional identity tensor and $\mu = \mu_{fluid}$ the dynamic viscosity. The strain rate tensor $\boldsymbol{\varepsilon}$ is defined by:

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \frac{1}{2}\left[\nabla \mathbf{v} + \nabla^{\mathrm{T}} \mathbf{v}\right] \tag{3.4}$$

In order to close the problem, Eq. (3.2) are subject to the homogeneous Dirichlet boundary conditions.

The weak form of problem (3.2) combined with (3.3) is obtained by multiplication of a test function and integration by parts. Let $\mathrm{H}^1(\Omega)$ be the Sobolev space of square integrable functions whose distributional derivatives are square integrable, and let $\mathrm{V} \subset \left[\mathrm{H}^1(\Omega)\right]^d$ be a functional space properly chosen according to the boundary conditions. Finally, let $\mathrm{Q} = \left\{q \in \mathrm{L}^2(\Omega) : \int_\Omega q = 0\right\}$. By denoting $(\cdot,\cdot)$ the scalar product of the space $\mathrm{L}^2(\Omega)$, the weak form of problem (3.2) on $\partial\Omega$ reads:

$$\begin{cases} \text{Find } (\mathbf{v}, p) \in \mathrm{V} \times \mathrm{Q} \text{ such that:} \\ \rho\ \left[(\partial_t \mathbf{v}, \mathbf{w}) + (\mathbf{v}\cdot\nabla\mathbf{v}, \mathbf{w})\right] + \left(2\mu\boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{w})\right) - \left(p, \nabla\cdot\mathbf{w}\right) = (\mathbf{f}, \mathbf{w}), \quad \forall \mathbf{w} \in \mathrm{V} \\ \left(\nabla\cdot\mathbf{v}, q\right) = 0, \quad \forall q \in \mathrm{Q}. \end{cases} \tag{3.5}$$

where ρ and μ are the density and the dynamic viscosity respectively.

The standard Galerkin approximation consists in decomposing the domain Ω into $\mathcal{N}$ elements Ω_e such that they cover the domain. Therefore, the elements are either disjoint or share a complete edge (or face in 3D). Using a partition $\mathcal{T}_h$, the above-defined functional spaces V and Q are approached by finite dimensional spaces V_h and Q_h such that:

$$\mathrm{V}_h = \{\mathbf{v}_h | \mathbf{v}_h \in \mathrm{C}^0(\Omega)^n, \mathbf{v}_{h|\Omega_e} \in \mathrm{P}^1(\Omega_e)^n,\ \forall \Omega_e \in \mathcal{T}_h\} \tag{3.6}$$

$$\mathrm{Q}_h = \{p_h | p_h \in \mathrm{C}^0(\Omega)^n, p_{h|\Omega_e} \in \mathrm{P}^1(\Omega_e)^n,\ \forall \Omega_e \in \mathcal{T}_h\} \tag{3.7}$$

The Galerkin discrete problem consists therefore in solving the following mixed problem:

$$\begin{cases} \text{Find } (\mathbf{v}_h, p_h) \in \mathrm{V}_h \times \mathrm{Q}_h \text{ such that:} \\ \rho\ \left[(\partial_t \mathbf{v}_h, \mathbf{w}_h) + (\mathbf{v}_h\cdot\nabla\mathbf{v}_h, \mathbf{w}_h)\right] + \left(2\mu\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)\right) - \left(p_h, \nabla\cdot\mathbf{w}_h\right) = (\mathbf{f}, \mathbf{w}_h), \quad \forall \mathbf{w}_h \in \mathrm{V}_h \\ \left(\nabla\cdot\mathbf{v}_h, q_h\right) = 0, \quad \forall q_h \in \mathrm{Q}_h. \end{cases} \tag{3.8}$$

It is well known that the stability of the semi-discrete formulation requires an appropriate choice of the finite element spaces V_h and Q_h that must fulfill a compatibility condition [42]. Accordingly, the standard Galerkin method using P1/P1 elements (i.e. the same piecewise linear space for V_h and Q_h) is not stable. Moreover, convection-dominant problems (i.e. problems where the convection term $\mathbf{v}\cdot\nabla\mathbf{v}$ is much larger than the diffusion term $\nabla\cdot\left(2\mu\ \boldsymbol{\varepsilon}\right)$) also lead to a loss of coercivity in the formulation (3.5). This phenomenon manifests itself as oscillations that pollute the solution.

3.2.2 The VMS formulation applied to the Navier-Stokes equations

In this work, we use a VMS method [40] which circumvents both the previously stated problems through a Galerkin approach. The basic idea is to consider that the unknowns can be split into two components, a coarse one and a fine one, corresponding to different scales or levels of resolution. First, we solve the fine scales in an approximate manner and then we replace their effect into the large-scale equation. We present here only an outline of the method, and the reader is referred to [41] for extensive details about the formulation.

3.2.2.1 Basic principles of the multiscale approach

Let us split the velocity and the pressure fields into resolvable coarse-scale and unresolved fine-scale components: $\mathbf{v} = \mathbf{v}_h + \mathbf{v}'$ and $p = p_h + p'$. The same decomposition can be applied to the weighting functions: $\mathbf{w} = \mathbf{w}_h + \mathbf{w}'$ and $q = q_h + q'$. Subscript h is used hereafter to denote the finite element (coarse) component, whereas the prime is used for the so called sub-grid scale (fine) component of the unknowns. The enrichment of the functional spaces is performed as follows: $\mathrm{V} = \mathrm{V}_h \oplus \mathrm{V}'$, $\mathrm{V}_0 = \mathrm{V}_{h,0} \oplus \mathrm{V}'_0$ and $\mathrm{Q} = \mathrm{Q}_h \oplus \mathrm{Q}'$. Thus, the finite element approximation for the time-dependent Navier-Stokes problem reads:

$$\left\{\begin{array}{l} \text{Find}(\mathbf{v}, p) \in \mathrm{V} \times \mathrm{Q} \text{ such that:} \\ \rho\left(\partial_t(\mathbf{v}_h + \mathbf{v}'), (\mathbf{w}_h + \mathbf{w}')\right)_\Omega + \rho\left((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), (\mathbf{w}_h + \mathbf{w}')\right)_\Omega + \left(2\mu\boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')\right)_\Omega \\ \qquad\qquad - \left((p_h + p'), \nabla \cdot (\mathbf{w}_h + \mathbf{w}')\right)_\Omega = \left(\mathbf{f}, (\mathbf{w}_h + \mathbf{w}')\right)_\Omega, \quad \forall \mathbf{w} \in \mathrm{V}_0 \\ \left(\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), (q_h + q')\right)_\Omega = 0, \quad \forall q \in \mathrm{Q}. \end{array}\right. \tag{3.9}$$

To derive the VMS formulation, we split Eq. (3.9) into a large-scale and a fine-scale problem. Integrating by parts within each element, we obtain the so-called coarse-scale problem:

$$\left\{\begin{array}{l} \rho\left(\partial_t(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h\right)_\Omega + \rho\left((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h\right)_\Omega + \left(2\mu\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)\right)_\Omega \\ \qquad\qquad - \left((p_h + p'), \nabla \cdot \mathbf{w}_h\right)_\Omega = (\mathbf{f}, \mathbf{w}_h)_\Omega, \quad \forall \mathbf{w}_h \in \mathrm{V}_{h,0} \\ \left(\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h\right)_\Omega = 0, \quad \forall q_h \in \mathrm{Q}_h. \end{array}\right. \tag{3.10}$$

and the so-called fine-scale problem:

$$\left\{\begin{array}{l} \rho\left(\partial_t(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}'\right)_{\Omega_e} + \rho\left((\mathbf{v}_h + \mathbf{v}') \cdot \nabla(\mathbf{v}_h + \mathbf{v}'), \mathbf{w}'\right)_{\Omega_e} + \left(2\mu\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}')\right)_{\Omega_e} \\ \qquad\qquad - \left((p_h + p'), \nabla \cdot \mathbf{w}'\right)_\Omega = \left(\mathbf{f}, \mathbf{w}'\right)_\Omega, \quad \forall \mathbf{w}' \in \mathrm{V}'_0 \\ \left(\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h\right)_\Omega = 0, \quad \forall q' \in \mathrm{Q}'. \end{array}\right. \tag{3.11}$$

where $(\cdot,\cdot)_\Omega$ represents the scalar product on the whole domain while $(\cdot,\cdot)_{\Omega_e}$ is the scalar product on element Ω_e.

To derive our stabilized formulation, we first solve the fine scale problem (3.11), defined on the sum of element interiors and written in terms of the time-dependent large scale variables. Then we substitute the fine scale solution back into the coarse problem (3.10), thereby *eliminating appearance of the fine-scale while still modelling their effects.* As in [41]; we recall here 3 important remarks/assumptions that have to be made:

- by considering the small scale velocity as bubble functions vanishing on the boundaries of the element, terms involving integrals over the element interior boundaries will be neglected,
- we neglect the second derivatives of the weighting function in the momentum residuals of (3.11),
- as the fine-scale space is assumed to be H^1-orthogonal to the finite element space, crossed viscous terms vanish in (3.10) and (3.11).

3.2.2.2 The fine scale sub-problem

Under several assumptions about the time-dependency and the non-linearity of the momentum equation of the sub-scale system detailed in [41], the fine-scale solutions $\mathbf{v}'$ and p' can be written in terms of the time-dependent large-scale variables using residual-based terms that are derived consistently. For all $\Omega_e \in \mathscr{T}_h$, we have:

$$\begin{array}{l} \mathbf{v}'|_{\Omega_e} = \tau_{\mathrm{K}}\mathscr{R}_{\mathrm{M}} \\ p' \approx \tau_{\mathrm{C}}\mathscr{R}_{\mathrm{C}} \end{array} \tag{3.12}$$

where the momentum residual $\mathscr{R}_{\mathrm{M}}$ and the continuity residual $\mathscr{R}_{\mathrm{C}}$ are expressed as:

$$\begin{array}{l} \mathscr{R}_{\mathrm{M}} = \mathbf{f} - \rho\partial_t\mathbf{v}_h - \rho\mathbf{v}_h \cdot \nabla\mathbf{v}_h - \nabla p_h \\ \mathscr{R}_{\mathrm{C}} = -\nabla \cdot \mathbf{v}_h \end{array} \tag{3.13}$$

In this work, we adopt the definition proposed by Codina in [43] for the stabilizing coefficient τ_{K} and τ_{C}:

$$\tau_{\mathrm{K}} = \left[\left(\frac{2\rho \|\mathbf{v}_h\|_{\Omega_e}}{h_e} \right)^2 + \left(\frac{4\mu}{h_e^2} \right)^2 \right]^{-\frac{1}{2}}, \tag{3.14}$$

$$\tau_{\mathrm{C}} = \left[\left(\frac{\mu}{\rho} \right)^2 + \left(\frac{c_2 \|\mathbf{v}_h\|_{\Omega_e}}{c_1 h_e} \right)^2 \right]^{\frac{1}{2}} \tag{3.15}$$

where h_e is the characteristic length of the element and c_1 and c_2 are algorithmic constants. We take them as $c_1 = 4$ and $c_2 = 2$ for linear elements. $\|\mathbf{v}_h\|_{\Omega_e}$ is the coarse scale velocity norm on the element, defined by:

$$\|\mathbf{v}_h\|_{\Omega_e} = \sqrt{\mathbf{v}_{x,h}^2 + \mathbf{v}_{y,h}^2 + \mathbf{v}_{z,h}^2} \tag{3.16}$$

3.2.2.3 The coarse scale sub-problem

Let us consider the coarse scale problem (3.10). Taking into account the assumptions prescribed in [41] and recalled in Section 3.2.2.1 for the fine scale fields, the large-scale system becomes:

$$\left\{ \begin{array}{l} \rho\,(\partial_t \mathbf{v}_h, \mathbf{w}_h)_\Omega + \left(\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h\right)_\Omega + \left(\rho \mathbf{v}_h \cdot \nabla \mathbf{v}', \mathbf{w}_h\right)_\Omega + \left(2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)\right)_\Omega \\ -\left(p_h, \nabla \cdot \mathbf{w}_h\right)_\Omega - \left(p', \nabla \cdot \mathbf{w}_h\right)_\Omega = (\mathbf{f}, \mathbf{w}_h)_\Omega, \quad \forall \mathbf{w}_h \in \mathrm{V}_{h,0} \\ \left(\nabla \cdot \mathbf{v}_h, q_h\right)_\Omega + \left(\nabla \cdot \mathbf{v}', q_h\right)_\Omega = 0, \quad \forall q_h \in \mathrm{Q}_h. \end{array} \right. \tag{3.17}$$

Then, integrating by parts the third term in the first equation and the second term in the second equation of (3.17) and substituting the expressions of both the fine-scale pressure and the fine-scale velocity of (3.12), the large-scale system reads:

$$\left\{ \begin{array}{l} \rho\,(\partial_t \mathbf{v}_h, \mathbf{w}_h)_\Omega + \left(\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h\right)_\Omega - \sum_{\Omega_e \in \mathcal{T}_h} \left(\tau_{\mathrm{K}} \mathcal{R}_{\mathrm{M}}, \rho \mathbf{v}_h \nabla \mathbf{w}_h\right)_{\Omega_e} + \left(2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)\right)_\Omega \\ \qquad -\left(p_h, \nabla \cdot \mathbf{w}_h\right)_\Omega - \sum_{\Omega_e \in \mathcal{T}_h} (\tau_{\mathrm{C}} \mathcal{R}_{\mathrm{C}}, \nabla \cdot \mathbf{w}_h)_{\Omega_e} = (\mathbf{f}, \mathbf{w}_h)_\Omega, \quad \forall \mathbf{w}_h \in \mathrm{V}_{h,0} \\ \left(\nabla \cdot \mathbf{v}_h, q_h\right)_\Omega - \sum_{\Omega_e \in \mathcal{T}_h} \left(\tau_{\mathrm{K}} \mathcal{R}_{\mathrm{M}}, \nabla q_h\right)_{\Omega_e} = 0, \quad \forall q_h \in \mathrm{Q}_h \end{array} \right. \tag{3.18}$$

Finally, substituing the residuals of the momentum equation and developping all the additional terms, we obtain a modified coarse scale formulation expressed exclusively in terms of coarse scale variables. The new modified problem for linear tetrahedral elements can now be decomposed into four main term: the first one is the standard Galerkin contribution, the second and the third terms take into account the influence of the fine-scale velocity on the finite element components and the last term models the influence of the fine-scale pressure onto the large-scale problem. We finally get:

$$\begin{array}{l} \rho\,(\partial_t \mathbf{v}_h + \mathbf{v}_h . \nabla \mathbf{v}_h, \mathbf{w}_h)_\Omega + \left(2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)\right)_\Omega - \left(p_h, \nabla . \mathbf{w}_h\right)_\Omega + \left(\nabla . \mathbf{v}_h, q_h\right)_\Omega - (\mathbf{f}, \mathbf{w}_h)_\Omega \\ + \displaystyle\sum_{\Omega_e \in \mathcal{T}_h} \tau_{\mathrm{K}} \left(\rho(\partial_t \mathbf{v}_h + \mathbf{v}_h . \nabla \mathbf{v}_h) + \nabla p_h - \mathbf{f}, \rho \mathbf{v}_h \nabla \mathbf{w}_h\right)_{\Omega_e} \\ + \displaystyle\sum_{\Omega_e \in \mathcal{T}_h} \tau_{\mathrm{K}} \left(\rho(\partial_t \mathbf{v}_h + \mathbf{v}_h . \nabla \mathbf{v}_h) + \nabla p_h - \mathbf{f}, \nabla q_h\right)_{\Omega_e} \\ + \displaystyle\sum_{\Omega_e \in \mathcal{T}_h} (\tau_{\mathrm{C}} \nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h)_{\Omega_e} = 0 \quad \forall \mathbf{w}_h \in \mathrm{V}_{h,0}\,, \quad \forall q_h \in \mathrm{Q}_h \end{array} \tag{3.19}$$

Compared to the standard Galerkin method, the proposed stable formulation involves additional integrals that are evaluated element-wise. These additional terms represent the stabilizing effect of the sub-grid scales and are introduced in a consistent way in the Galerkin formulation. They make it possible to avoid instabilities caused by both dominant convection terms and incompatible approximation spaces.

3.3 Convection-diffusion equation

3.3.1 Governing equation

The second PDE system we solve is the thermal convection-diffusion system. To begin this section, let's introduce some notations. Referring to Irissarri et al. in [44], we express the PDE as:

$$\begin{cases} \partial_t u - a\Delta u + \mathbf{v}\cdot\nabla u = f \quad in\,\Omega \\ u = g \quad on\,\Gamma_g \\ \mathscr{B}u = h \quad on\,\Gamma_h \end{cases} \tag{3.20}$$

where ∇ and Δ are the gradient and the Laplacian operator, a is the diffusion coefficient and $\mathbf{v}$ the velocity field. $\mathscr{B}$ is an operator acting on the boundary that defines the natural boundary condition; g is the value of the Dirichlet boundary condition and h determines the Neumann boundary condition. In this work, we consider only Dirichlet boundary conditions. For the case of a convection-diffusion equation, we define the generic differential operator:

$$\mathscr{L}u = -a\Delta u + \mathbf{v}\cdot\nabla u \tag{3.21}$$

We define the solution and test function spaces as standard Sobolev spaces:

$$\begin{aligned} \mathscr{S} &= \{u \in \mathrm{H}^1(\Omega) \mid u = g\ on\,\Gamma_g\} \\ \mathscr{V} &= \{w \in \mathrm{H}^1(\Omega) \mid w = 0\ on\,\Gamma_g\} \end{aligned} \tag{3.22}$$

The variational formulation of Eq. (3.20) is:

$$\begin{cases} \text{Find } u \in \mathscr{S} \text{ such that:} \\ \\ a(w,u) = (w,f), \quad \forall w \in \mathscr{V} \end{cases} \tag{3.23}$$

where $a(\cdot,\cdot)$ is a bilinear form, $(\cdot,\cdot)$ the $\mathrm{L}^2(\Omega)$ inner product.

3.3.2 The Galerkin FE formulation

Applying the Finite Element Method (FEM), we mesh the domain into $\mathscr{N}$ non-overlapping elements Ω_e. We write $\tilde{\Omega}$ and $\tilde{\Gamma}$ as:

$$\tilde{\Omega} = \bigcup_{e=1}^{\mathscr{N}} \Omega_e \quad \tilde{\Gamma} = \bigcup_{e=1}^{\mathscr{N}} \Gamma_e \setminus \Gamma \tag{3.24}$$

We define the FE solution and weighting function spaces $\mathscr{S}^h \subset \mathscr{S}$ and $\mathscr{V}^h \subset \mathscr{V}$ such that:

$$\begin{aligned} \mathscr{S}^h &= \{u_h \in \mathrm{H}^1(\Omega) \mid u_{h|\Omega_e} \in \mathbb{P}_k,\ u_{h|\Gamma_g} = g\ \forall \Omega_e \in \tilde{\Omega}\} \\ \mathscr{V}^h &= \{w_h \in \mathrm{H}^1(\Omega) \mid w_{h|\Omega_e} \in \mathbb{P}_k,\ w_{h|\Gamma_g} = 0\ \forall \Omega_e \in \tilde{\Omega}\} \end{aligned} \tag{3.25}$$

with $\mathbb{P}_k$ the space of polynomial of degree k. In this work, we will only use linear elements, hence $k = 1$.

With these definitions, we apply the standard Galerkin method:

$$\begin{cases} \text{Find } u_h \in \mathscr{S}^h \text{ such that,} \\ \\ a(w_h,u_h) = (w_h,f), \quad \forall w_h \in \mathscr{V}^h \end{cases} \tag{3.26}$$

It is well known that this formulation is unstable and leads to spurious oscillations when the convective term of the equation is dominant. For this reason, we stabilize the formulation using the SUPG numerical scheme.

3.3.3 Streamline Upwind Petrov-Galerkin (SUPG) method

The SUPG method consists in introducing stabilizing terms in the standard Galerkin formulation. To do so, for all terms in Eq. (3.26), we replace the weighting function w_h by a new weighting function $w_h + \tau_e \mathbf{v} \cdot \nabla w_h$. This modification of the formulation is usually interpreted as adding more weight to the node upstream, reducing the weight on the node downstream. It adds an artificial weighted diffusion along the streamline direction. This stabilization is done locally and on each element we have:

$$a(w_h + \tau_e \mathbf{v} \cdot \nabla w_h, u_h) = (w_h + \tau_e \mathbf{v} \cdot \nabla w_h, f) \ \ \forall w_h \in \mathcal{V}^h \tag{3.27}$$

Thanks to the bi-linearity of $a(\cdot,\cdot)$ and of $(\cdot,\cdot)$, the formulation can be written as:

$$a(w_h, u_h) + a(\tau_e \mathbf{v} \cdot \nabla w_h, u_h) - (\tau_e \mathbf{v} \cdot \nabla w_h, f) = (w_h, f) \ \ \forall w_h \in \mathcal{V}^h \tag{3.28}$$

And so,

$$a(w_h, u_h) + (\tau_e \mathbf{v} \cdot \nabla w_h, \mathscr{L} u_h - f) = (w_h, f) \ \ \forall w_h \in \mathcal{V}^h \tag{3.29}$$

Finally, summing on each element, we have a new bi-linear form $a_\tau(\cdot,\cdot)$ and we can write the following formulation:

$$a_\tau(w_h, u_h) = a(w_h, u_h) + \sum_{\Omega_e \in \tilde{\Omega}} (\mathbf{v} \cdot \nabla w_h, \tau_e(\mathscr{L} u_h - f))_{\Omega_e} = (w_h, f) \ \ \forall w_h \in \mathcal{V}^h \tag{3.30}$$

Concerning the choice of the stabilizing parameter τ_e, we refer to Hughes et al. in [45] and we choose:

$$\tau_e = \frac{h_e}{2||\mathbf{v}||_2} \left(coth(\mathrm{Pe}_{\Omega_e}) - \frac{1}{\mathrm{Pe}_{\Omega_e}} \right) \tag{3.31}$$

where h_e is the local mesh size, computed as described in [46]. Pe_{Ω_e} is the local Peclet number defined as:

$$\mathrm{Pe}_{\Omega_e} = \frac{||\mathbf{v}||_2 h_e}{2a} \tag{3.32}$$

Remark 2 *In order to study the error of this numerical scheme, we use the VMS analysis. In fact, it has to be noted that the SUPG scheme can be considered as a particular form of the generalized VMS formulation. Indeed, the stabilizing term of Eq. (3.30) can also be seen as the effect of the subscales on the coarse scales. Hughes gives more details about this concordance in [40].*

3.3.4 The VMS formulation applied to the Convection-Diffusion equation

As in the previous section, the VMS formulation consists in decomposing the solution and test functions spaces into two sub-spaces: a mesh scale subspace (or coarse scales) $(\mathscr{S}^h, \mathcal{V}^h)$ and an under-mesh scale subspace (or subscales) $(\mathscr{S}', \mathcal{V}')$ such that $\mathscr{S} = \mathscr{S}^h \oplus \mathscr{S}'$ and $\mathcal{V} = \mathcal{V}^h \oplus \mathcal{V}'$. Therefore, we can decompose the solution and test functions as follow:

$$\begin{aligned} u = u_h + u', \ \ u_h \in \mathscr{S}_h, \ \ u' \in \mathscr{S}' \\ w = w_h + w', \ \ w_h \in \mathscr{W}_h, \ \ w' \in \mathscr{S}' \end{aligned} \tag{3.33}$$

Thanks to the orthogonality between the coarse scales subspace and the subscales subspace, the variational form can be split into a coarse scales sub-problem and a subscales sub-problem [45]:

$$\begin{aligned} a(w_h, u_h) + a(w_h, u') = (w_h, f) \ \ \forall w_h \in \mathcal{V}^h \\ a(w', u_h) + a(w', u') = (w', f) \ \ \forall w' \in \mathcal{V}' \end{aligned} \tag{3.34}$$

We start by solving the subscales sub-problem (second equation). For smooth functions on the element interior but rough across the inter-element boundaries, the integration by parts leads to the following equation:

$$\begin{aligned} a(w', u') &= -a(w', u_h) + (w', f) \\ a(w', u') &= -(w', \mathscr{L} u_h - f) - (w', [\mathscr{B} u_h])_{\tilde{\Gamma}} - (w', \mathscr{B} u_h)_{\Gamma_h} \ \ \forall w' \in \mathcal{V}' \end{aligned} \tag{3.35}$$

where the jump term $[\cdot]$ represents the difference of the fluxes on both sides of the element boundaries (see [47] for more details). An analytic solution of problem (3.35) can be found. This solution will be developed in the next chapter.

3.4 Aerothermal coupling in turbulent flows

As mentioned in the introduction of this chapter, we solve decoupled aerothermal problems in Cimlib-CFD. Therefore, we have to implement additional numerical models in order to take into account for the aerothermal coupling between the temperature u and the velocity/pressure couple $(\mathbf{v}, p)$. To do so, we compute both: (i) a modified dynamic viscosity μ_{total} and (ii) a modified thermal conductivity λ_{total}.

Indeed, a thermal model is first used to compute the modified fluid dynamic viscosity μ_{T} which will then be taken as input in the Navier-Stokes VMS solver. The modified fluid dynamic viscosity μ_{T} takes into account the variations of dynamic viscosity due to the temperature's fluctuations in the fluid. To compute it, we use the well known Sutherland law introduced in [48].

Then, a sub-grid scale turbulence model is used to compute the turbulent viscosity μ_{turb}; which will be then used to compute the modified thermal conductivity λ_{total}; which will then be taken as an input for the thermal convection-diffusion solver. The turbulent viscosity μ_{turb} allows taking into account the turbulent characteristics of the flow in the computation of the modified thermal conductivity λ_{total}. To compute it, we use the well known Smagorinsky Model (SM) introduced in [49].

3.4.1 Thermal model

This first numerical model is used to compute the modified fluid dynamic viscosity μ_{T} that takes into account the dynamic viscosity variations due to temperature's fluctuations in the fluid. Indeed, Sutherland in [48] explicit the modification of fluid viscosity along with its temperature with the following formula:

The Sutherland's law [48]:

$$\mu_{\mathrm{T}} = \mu_{fluid} \left(\frac{\mathrm{T}}{\mathrm{T}_{init}} \right)^{3/2} \frac{\mathrm{T}_{init} + \mathrm{C}}{\mathrm{T} + \mathrm{C}} \tag{3.36}$$

with μ_{fluid} the tabulated fluid dynamic viscosity at T_{init} and C a fixed constant equals to 110.4 K.

3.4.2 Sub-grid scale turbulence model

The second numerical model is used to compute the turbulent dynamic viscosity μ_{turb} which allows taking into account the turbulent characteristics of the fluid in the modified thermal conductivity λ_{total}. To compute the turbulent dynamic viscosity μ_{turb}, we use the well known SM which is written as follows:

The Smagorinsky Model (SM) [49]:

$$\mu_{turb} = \rho_{fluid} (\mathrm{C}_s \Delta_e)^2 |\bar{\mathrm{S}}_{\mathbf{v}}| = \rho_{fluid} (\mathrm{C}_s \Delta_e)^2 \sqrt{2 \widehat{\mathrm{S}_{ij}} \widehat{\mathrm{S}_{ij}}} \tag{3.37}$$

where ρ_{fluid} is the tabulated fluid density, Δ_e is the length scale related to the grid size. We take Δ_e as the cubic root of the cell volume. Therefore we have: $\Delta_e = |\Omega_e|^{1/3}$ with $|\Omega_e|$ the volume of the element Ω_e. One can note that this definition works well on isotropic elements but can be discussed in the case of anisotropic meshes. The term $\mathrm{S}_{\mathbf{v}}$ corresponds to the symmetric part of the velocity gradient and, $\widehat{\mathrm{S}_{ij}}$ corresponds to the symmetric part of the resolved velocity gradient on the mesh nodes.

The Smagorinsky constant C_s needs to be specified prior to the simulation. The lack of clarity concerning the optimal value of the Smagorinsky constant is one of the main drawback of this model. Indeed, the Smagorinsky constant C_s depends on: (i) the type of flow, (ii) the filter being used and (iii) the numerical method. In [50], Lilly et al. found a theoretical value equals to 0.18 whereas Germano et al. in [51] have introduced and developed a dynamic sub-grid scale model. Indeed, the authors replaced the constant coefficient C_s by a parameter $\mathrm{C}_s(x,t)$ which evolves dynamically in space and time. In this study, we choose the theoretical value of Lilly et al. and we take $\mathrm{C}_s = 0.18$.

The second drawback concerns the introduction of too much numerical diffusion in particular in the laminar regions located near the plate. In order to reduce the turbulent viscosity near the walls and to take into account the anisotropy of the turbulence, the SM is modified using the Van Driest damping function introduced in [52]. We have:

$$\mu_{turb} = \rho_{fluid}(\mathrm{C}_s\Delta_e f_\mu)^2|\bar{\mathrm{S}}| \tag{3.38}$$

with

The Van Driest damping function:

$$f_\mu = 1 - e^{-z^+/25} \tag{3.39}$$

where z^+ is the dimensionless wall distance. Indeed, z^+ is defined by (i) the absolute distance to the nearest wall z, (ii) the friction velocity at the nearest wall u_τ and (iii) the local fluid kinematic viscosity ν_{fluid}. It is written as follow:

$$z^+ = \frac{z\,u_\tau}{\nu_{fluid}} \tag{3.40}$$

where u_τ is computed thanks to the wall shear stress τ_w. We have:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho_{fluid}}} \tag{3.41}$$

Therefore, we get a new expression of z^+ only written in terms of fluid dynamic viscosity:

$$z^+ = \frac{\rho_{fluid}\,z}{\mu_{fluid}}\sqrt{\frac{\tau_w}{\rho_{fluid}}} \tag{3.42}$$

In Equation 3.42, the wall shear stress τ_w is given by:

$$\tau_w = \mu_{fluid}\,(\nabla\mathbf{v}\cdot\mathbf{n})_{\alpha=0} \tag{3.43}$$

where $\mathbf{n}$ is the normal direction of the walls. To compute the normal $\mathbf{n}$, we use the Level-Set function α characterizing the walls. Using the Level-Set method, we can define the normal direction to the walls as:

$$\mathbf{n} = \frac{\nabla\alpha}{||\nabla\alpha||} \tag{3.44}$$

Finally, we define a new dynamic viscosity μ_{total} that takes into account (i) the Sutherland law and (ii) the turbulent sub-grid scale model. We have:

$$\mu_{total} = \mu_{\mathrm{T}} + \mu_{turb} \tag{3.45}$$

3.4.3 Effects of turbulence on thermal conductivity

Now that we have computed the turbulent dynamic viscosity μ_{turb}, we compute a modified thermal conductivity λ_{total} that takes into account the turbulent effects of μ_{turb}. It is written as:

$$\lambda_{total} = \lambda_{fluid} + \lambda_{turb} \tag{3.46}$$

where the first term λ_{fluid} is the tabulated thermal fluid conductivity.

The second term however, allows taking into account the change of conductivity due to the flow turbulence. It is written λ_{turb} and is expressed with the following formula:

$$\lambda_{turb} = \frac{c_p\ \mu_{turb}}{\mathrm{P}r} \tag{3.47}$$

with c_p the specific heat of the fluid and $\mathrm{P}r$ the Prandtl number defined here with the Rayleigh number $\mathrm{R}a$ and the Grashof number $\mathrm{G}r$ according to the following definition:

$$\mathrm{P}r = \frac{\mathrm{R}a}{\mathrm{G}r} \tag{3.48}$$

The Rayleigh number is defined by the following formula:

$$\mathrm{R}a = \frac{g\beta}{\nu_{fluid}\alpha_{fluid}}(\mathrm{T}_{wall} - \mathrm{T}_{ref})\mathrm{L}_c^3 \tag{3.49}$$

where g is the acceleration due to gravity, β is the thermal expansion coefficient, ν_{fluid} is the kinematic viscosity, α_{fluid} is the thermal diffusivity, T_{wall} is the temperature of the hot wall, T_{ref} is the temperature far from the wall and L_c is the characteristic length of the problem.

Using also these parameters, the Grashof number is defined by the following expression:

$$\mathrm{G}r = \frac{g\beta(\mathrm{T}_{wall} - \mathrm{T}_{ref})\mathrm{L}_c^3\rho_{fluid}^2}{\mu_{fluid}} \tag{3.50}$$

Therefore, we can deduce the Prandtl number expression which is:

$$\mathrm{P}r = \frac{\nu_{fluid}}{\alpha_{fluid}} \tag{3.51}$$

Finally, we compute the modified thermal conductivity with the following equation:

$$\lambda_{total} = \lambda_{fluid} + \frac{c_p\ \mu_{turb}}{\mathrm{P}r} \tag{3.52}$$

To conclude, we indeed achieve a strong correlation between the turbulent activity and the fluid's thermal properties. The proposed coupling allows to take into account: (i) the modifications of the dynamic viscosity due to the temperature's variations and (ii) the modifications of the fluid thermal conductivity due to the sub-grid scale turbulence. Thus, we reinforce the aerothermal coupling of the simulation.

3.5 Validation case: mixed convection problem of a 3D ventilated cavity

In this section, we propose to validate the numerical methods previously introduced on a well known aerothermal benchmark called the 3D ventilated cavity. The validation case presented here concerns the numerical simulation of an indoor airflow inside an empty room equipped with a mechanical ventilation system. In this aerothermal problem, the airflow is moved around the room by mixed convection. Indeed, the air distribution in the room results of two effects: (i) the effect due to the temperature difference inside the room (natural convection) and (ii) the effect of a mechanical ventilation system (forced convection).

In the past years, this case has been treated with the two common approaches: the experimental approach and the numerical approach. The experimental approach requires real life measurement in a full-scale environmental chamber that needs to be isolated from the external world. The work of Nielsen et al. in [53] proposes an experimental study with measurement, models and calculations of the velocity and temperature characteristics in two dimensions. This work provides a consistent database of velocity and temperature distributions that has been largely used in the literature and allows numerical scientists to compare their results to real-life experimental measurements.

From a numerical science point of view, indoor airflows have been extensively investigated in the past decades. In particular, they have proved to be reliable benchmarks for numerous turbulence model adjustments. Therefore, several works can be found in the literature concerning simulations of indoor airflows using RANS or LES approaches. In particular, Chen et al. use this benchmark to asses the performance of $k-\epsilon$ models in [54]. Furthermore, Zhang et al. in [55] investigate two LES models: a classical SM, and a Filtered Dynamic Scale Model (FDSM). Their results show that the predicted velocity, temperature and turbulence distribution agree reasonably well with the experimental data. However, they observe that the FDSM has trouble determining with good accuracy the turbulence distribution and therefore, the heat transfer inside the cavity. The SM is therefore said to be preferred in this configuration.

The goal of this study is to validate the numerical methods presented in this chapter in term of (i) airflow solution, (ii) heat transfer and (iii) aerothermal coupling. To do so, in Section 3.5.1, we start by presenting the geometry and boundary conditions of the case along with the description of the 3 different computations realized to validate the aerothermal coupling. Then, in Section 3.5.2.1 the results in terms of global averaged velocity and temperature fields are presented and analyzed. Furthermore, the different resulting variables related to the aerothermal coupling are presented and discussed in Section 3.5.2.2. Finally, a comparison between the three aerothermal computations and the experimental data is made in Section 3.5.2.3, leading to validation of the numerical framework.

3.5.1 Description

3.5.1.1 Geometry and boundary conditions

The geometry of the cavity is presented in Figure 3.1. The cold jet is injected through a slot of height $h_{in} = 0.018\ m$ with a velocity magnitude $\mathrm{V}_{in} = 0.57\ m.s^{-1}$. These two parameters give a Reynolds number value at the injection equals to 684. The temperature of the cold jet is fixed at $\mathrm{T}_{ref} = 15°\mathrm{C}$. The two walls of normal X axis, along with the top wall are maintained at temperature T_{ref}. The wall corresponding to the heated floor is maintained at temperature $\mathrm{T}_{wall} = 35°\mathrm{C}$. Finally, the two walls of normal Y axis are considered to be adiabatic. The airflow is free to evacuate the cavity through an output slot of height $h_{out} = 0.024\ m$. At initialization, except on the boundary conditions, the velocity field is null everywhere inside the cavity. The geometric and physical parameters of this particular configuration are given in Table 3.1.

Except for the walls, the initial temperature in the domain is taken as the averaged value between the temperature of the heated wall T_{wall} and the temperature of the injected jet T_{ref}. We have:

$$\mathrm{T}_0 = \mathrm{T}_{mean} = \frac{\mathrm{T}_{wall} + \mathrm{T}_{ref}}{2} = 25°\mathrm{C} \tag{3.53}$$

Form Eq. 3.49, and replacing by the values given in Table 3.1, we compute the Rayleigh number and we obtain $\mathrm{R}a = 2.4e^{9}$. As well, from Eq. 3.50, we obtain the Grashof number $\mathrm{G}r = 3e^{9}$.

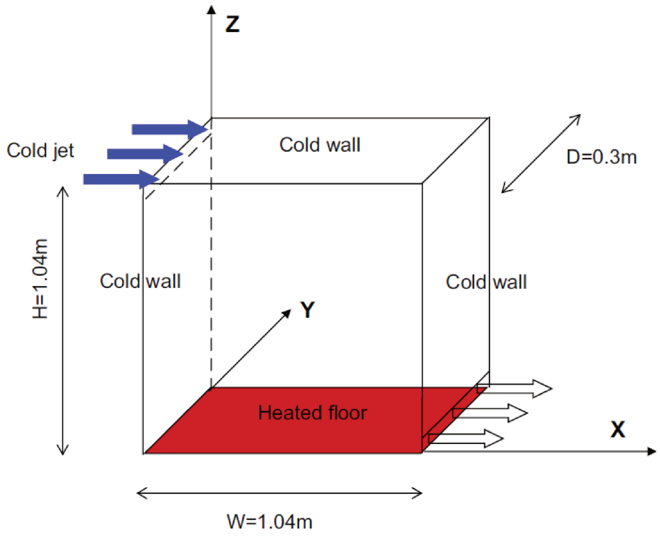


Figure 3.1 – Schematic view of the 3D ventilated cavity from Ezzouhri et al. in [56].

Parameter	Numerical value
g	9.81 $m.s^{-2}$
β	$3.4e^{-3}$ K^{-1}
ν_{fluid}	$15.6e^{-6}$ $m^2.s^{-1}$
μ_{fluid}	$19e^{-6}$ $\mathrm{P}a.s$
ρ_{fluid}	$15.6e^{-6}$ $kg.m^{-3}$
λ_{fluid}	0.027 $\mathrm{W}.m^{-1}.\mathrm{K}^{-1}$
α_{fluid}	$20e^{-6}$ $m^2.s^{-1}$
c_p	1004 $\mathrm{J}.kq^{-1}.\mathrm{K}^{-1}$
T_{wall}	308.15 K
T_{ref}	288.15 K
T_0	298.15K
L_c	1.04 m

Table 3.1 – 3D ventilated cavity parameters at 15°C.

Finally, from Eq. 3.51, we deduce the Prandtl number $\mathrm{P}r = \frac{2.4e^9}{3e^9} = 0.8$. These results on thermal characteristics are confirmed by the work of Zhang et al. in [55] and Ezzouhri et al. in [56].

3.5.1.2 Numerical procedure

According to Zhang et al. in [55], in order to reach a suitable heat transfer regime inside the cavity and to be able to confront the results with experimental data, the simulation duration t_{total} needs to be around 120 s. After several experiments, we choose a time step Δt equals to 0.01 s giving a number of increments equals to 12 000.

The mesh used here is an isotropic unstructured mesh. It is composed by 255 881 points and 1 405 903 P1 elements. It has been refined prior to the simulation to obtain the resulting mesh presented in Figure 3.2. Following the recommendations from François et al. in [57], the mesh is refined on the walls of normal X axis and Z axis where the heat transfer are expected to occur. Refining the mesh on these walls also allows for the input and output velocity to be well resolved on the geometry.

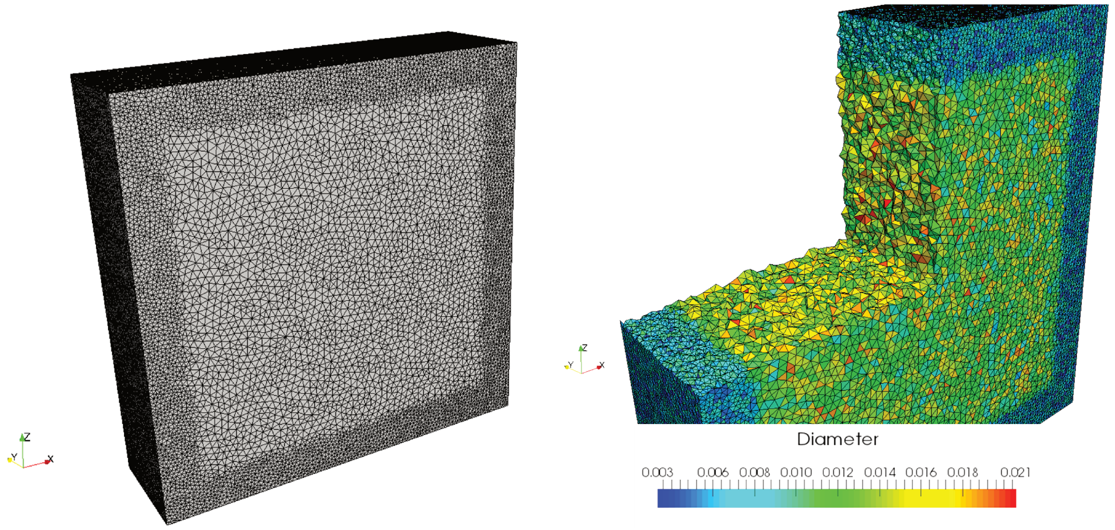


Figure 3.2 – Global mesh of the 3D ventilated cavity (left) and elements diameter distribution (right).

3.5.1.3 Choices of sub-grid scale modeling for aerothermal coupling

In this study, we propose to compare three different computations to analyze the effects of the two implemented numerical models used to achieve aerothermal coupling. To do so, we thus propose to use the Navier-Stokes VMS solver coupled with a thermal convection-diffusion solver with different input parameters in terms of (i) dynamic viscosity and (ii) thermal conductivity. The three different computations are summarized in Table 3.2.

Computation A corresponds to an aerothermal solution computed without considering any of the two additional numerical models for the fluid dynamic viscosity. As well, the thermal conductivity is computed without considering the effects of turbulence. On the contrary, *Computation B* uses both (i) the Sutherland law and (ii) the turbulence sub-grid scale model to compute both the dynamic viscosity and the thermal conductivity. The aerothermal coupling is therefore stronger than the two other computations. To finish, *Computation C* uses only the Sutherland law for the dynamic viscosity but keeps the turbulent dynamic viscosity μ_{turb} to compute the thermal conductivity. This way, the coupling is reinforced only by considering the dynamic viscosity variations due to the temperature and by considering the flow turbulence only in the thermal conductivity.

Solver	Computation A	Computation B	Computation C
Navier-Stokes VMS	μ_{fluid}	μ_{total}	μ_T
Convection-Diffusion	λ_{fluid}	λ_{total}	λ_{total}

Table 3.2 – Three types of numerical modeling for the aerothermal coupling

In the next section, we will compare the three computations A, B and C with experimental data and we will deduce which one is the best choice to solve this kind of aerothermal problems in Cimlib-CFD.

3.5.2 Results on the 3D ventilated cavity

In this section, we first present the results in term of temporal averaged velocity and temperature fields to asses for the qualitative shape of the aerodynamic and thermal solutions. The presented computation corresponds to *Computation B* which requires the computation of all the aerothermal coupling parameters. It allows us then to present the variations of the aerothermal coupling parameters inside the 3D ventilated cavity. Finally, a comparison with experimental data and previous numerical studies is proposed to validate the results.

3.5.2.1 Resulting temporal averaged velocity and temperature fields

The results on velocity, pressure and temperature fields presented in this section are the resulting temporal averaged fields computed over the complete simulation (i.e. during $t_{total} = 120\ s$). The temporal averaged values are computed at each point of the mesh taking into account a data output frequency of 100 increments.

As a start, we present in Figure 3.3 a slice view of the mean velocity magnitude field along with the streamlines of the mean velocity vector inside the cavity. We can observe the characteristic clockwise movement of the fluid due to the injection of air at the top left corner. Also, as shown by Ezzouhri et al. in [56], we observe on the streamline figure the counter clockwise movement of the fluid in the upper right part and the bottom left part of the cavity.

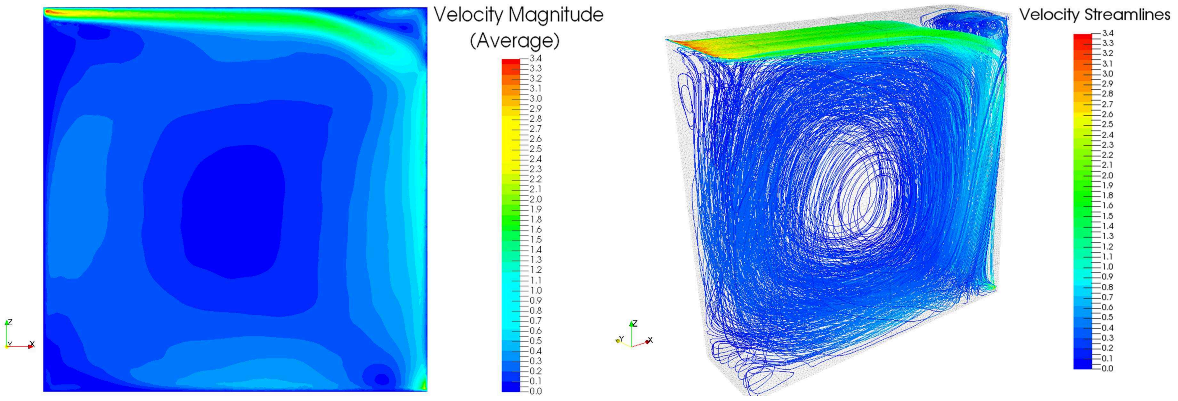


Figure 3.3 – Slice view of the mean velocity magnitude on plan XZ at Y = 0.15 (left) and streamlines of the mean velocity vector (right).

The mean temperature field is presented in Figure 3.4. We observe that the heat from the hot plate is well convected inside the cavity and follows the clockwise movement of the fluid.

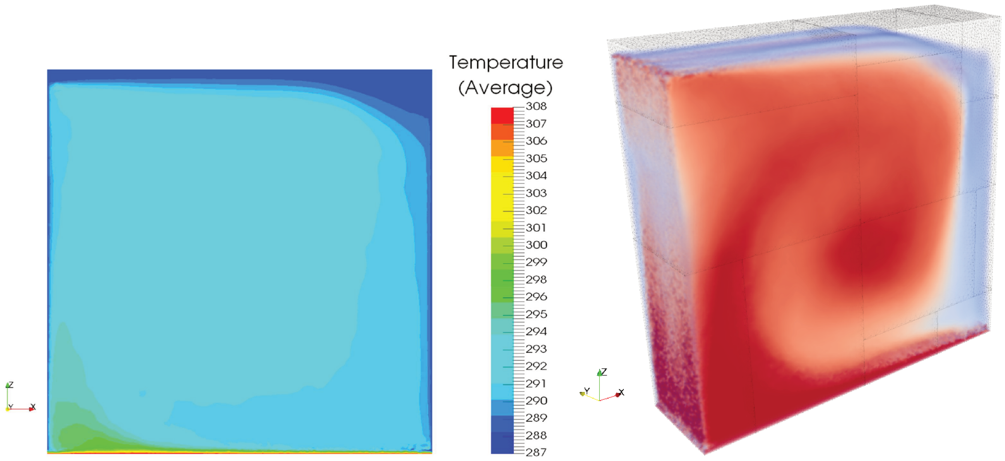


Figure 3.4 – Slice view of the mean temperature field on plan XZ at Y = 0.15 (left) and volume view of the mean temperature field (right).

3.5.2.2 Resulting aerothermal coupling parameters

The results on the aerothermal coupling parameters computed with the Sutherland law and the turbulence sub-grid scale model described in Section 3.4 are presented hereafter. For the 3D ventilated cavity, the level set is defined on the walls of normal X axis and Z axis. The results on the Level-Set function α and the resulting normal **n** are presented in Figure 3.5. On these figures, we observe that the Level-Set function is correctly computed according to each wall. The computation of the normal using the Level-Set method is also validated in this figure.

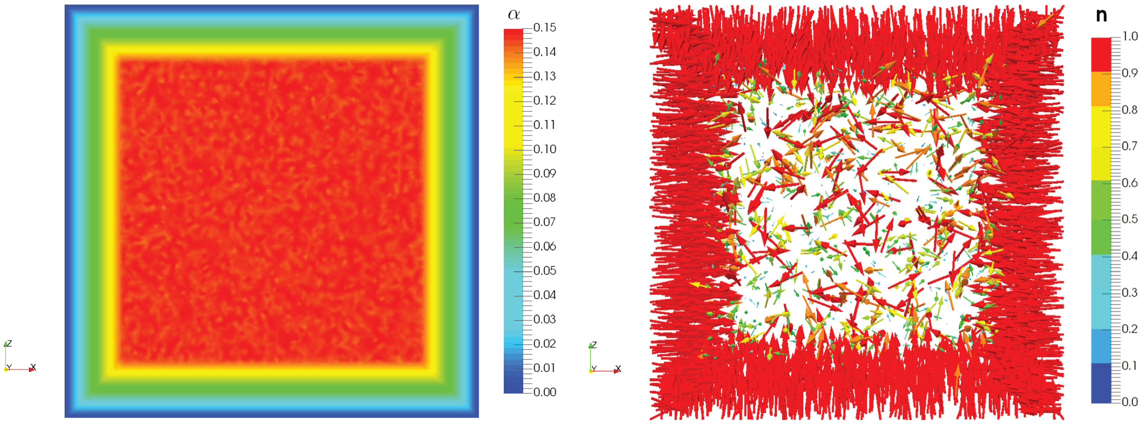


Figure 3.5 – The *Level-Set* function α (left) and the normal **n** to the walls (right).

Then, the results on the Sutherland dynamic viscosity μ_T and the wall shear stress τ_w are presented in Figure 3.6. As expected, the Sutherland dynamic viscosity follows the averaged temperature field and varies in a range of $1e^{-6}$ representing 5 % of the tabulated value. The wall shear stresses τ_w are computed following the method described in Section 3.4.2. As expected, we observe large values of τ_w near the walls and where significant velocity gradients occur.

From the computation of the wall shear stresses, we deduce the dimensionless wall distance z^+ and the associated Van Driest damping function f_μ. They are presented in Figure 3.7. We observe that the dimensionless wall distance reaches its maximum at the top right corner of the cavity where the velocity gradients are considerable due to the changes in flow direction and due the previously mentioned re-circulations that occur in this specific area. Following this results, the

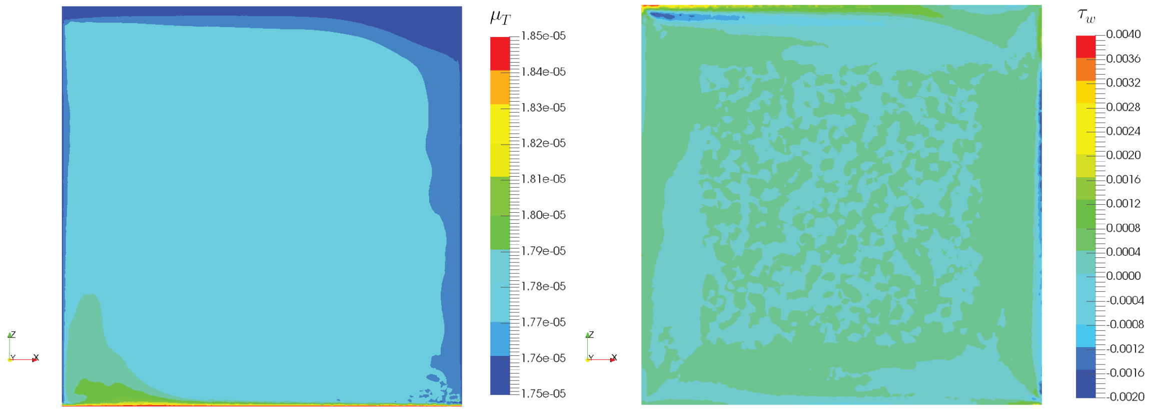


Figure 3.6 – The Sutherland dynamic viscosity μ_T (left) and the wall shear stress τ_w (right).

shape of the Van Driest damping function is in good agreement with our expectations. It reaches zero on the walls and a maximum value where z^+ is maximum.

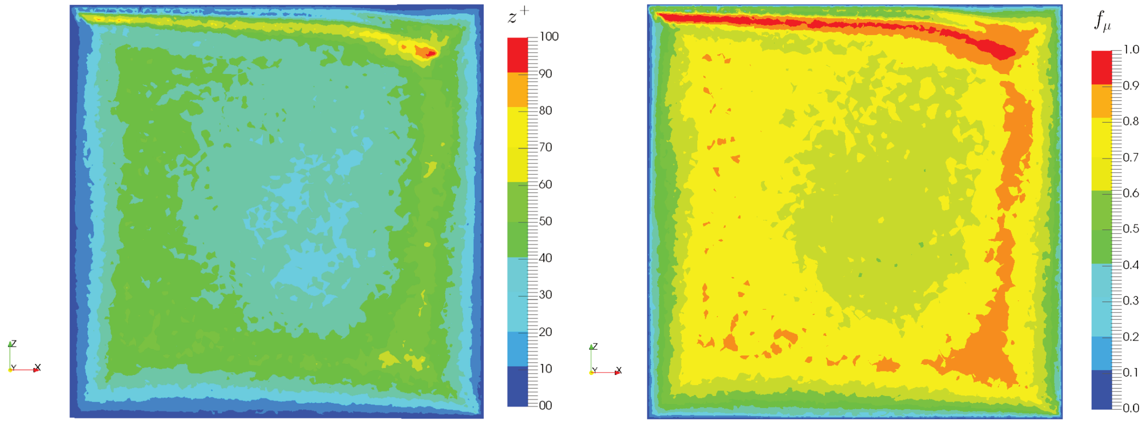


Figure 3.7 – The dimensionless wall distance z^+ (left) and the Van Driest damping function f_μ (right).

Finally, the resulting computations of the total dynamic viscosity μ_{total} and the total thermal conductivity λ_{total} are presented in Figure 3.8. On these figures, we observe that these two aerothermal coupling parameters reach large values where the turbulence modeling is significant. We can thus observe that the Sutherland law contribution μ_T is negligible and that the major contribution to the total dynamic viscosity comes from the sub-grid scale turbulence model contribution μ_{turb}. Consequently, the thermal conductivity experienced large variations in the same areas that the dynamic viscosity. Finally, we observe that for the 3D ventilated cavity, the contribution of turbulence modeling in the aerothermal coupling is, indeed, non-negligible.

3.5.2.3 Comparison with experimental and numerical studies

In order to validate the two additional models used for the aerothermal coupling in this chapter, we compare, in this section, the 3 computations with experimental data and a previous numerical study. To do so, we set sensors inside the cavity at $x = 0.52\ m$, $y = 0.15\ m$ and every $0.05\ m$ along the Z axis direction. At every increments and on each sensor, we extract two different data: (i) the x component of the velocity (named U) and (ii) the temperature T. Then, we compute a temporal averaged value of theses two data and we compare our results with: (i) experimental data from Nielsen et al. in [53] and (ii) numerical results from Zhang et al in [55].

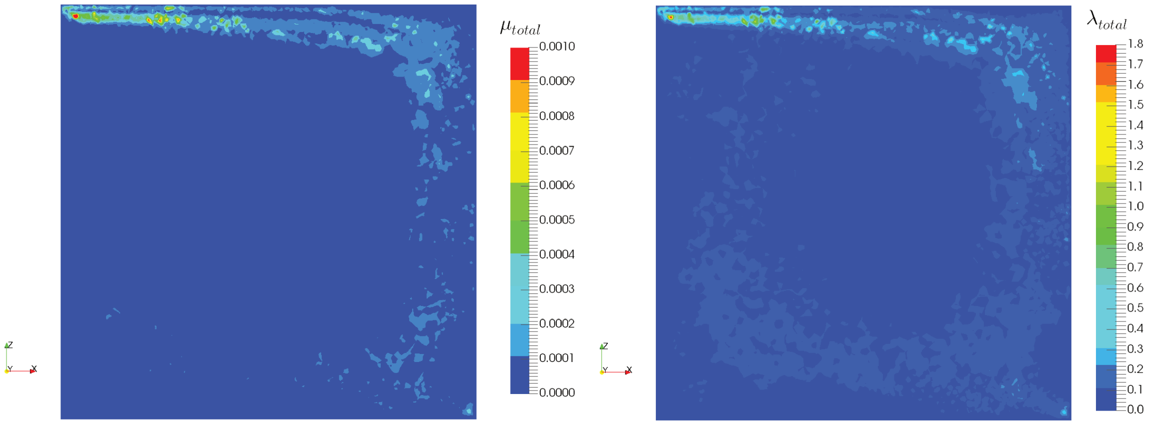


Figure 3.8 – The total dynamic viscosity μ_{total} (left) and the total thermal conductivity λ_{total} (right).

The results concerning the velocity field are presented in Figure 3.9. On this figure, we observe that the three computations are in general good agreement with the experimental and numerical references. However, we notice that there is an overestimation of the velocity component U when we get closer to the extremity of the curve. Furthermore, we observe that the best of the 3 computation is *Computation B* corresponding to the strongest aerothermal coupling. In fact, we observe that we get slightly closer to the references with this computation.

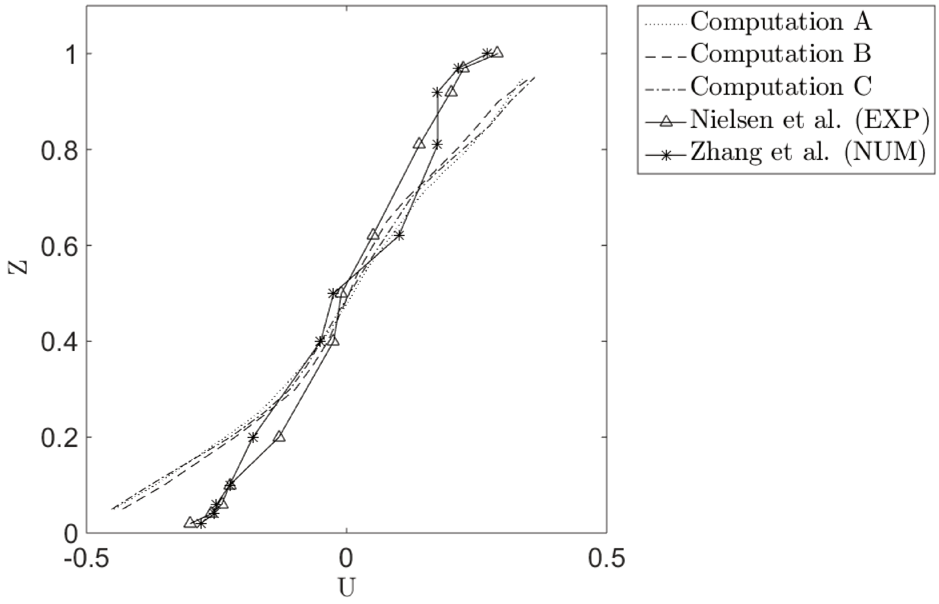


Figure 3.9 – Results on the x component of velocity at $x = 0.52\ m$, $y = 0.15\ m$ along the Z axis direction.

The results concerning the temperature along the Z axis direction are given in Figure 3.10. On this figure, we observe again that the three computations are in good agreement with the experimental study. Furthermore, these results are in better agreement than the numerical results from Zhang et al.. In general, we can say that the temperature inside the cavity is very well estimated with our numerical methods. If we look closer (see Figure 3.11), we can see that, referring to experimental results, *Computation B* gives the best results compared to the other computations.

From the comparison with experimental data, the best choice for aerothermal coupling is

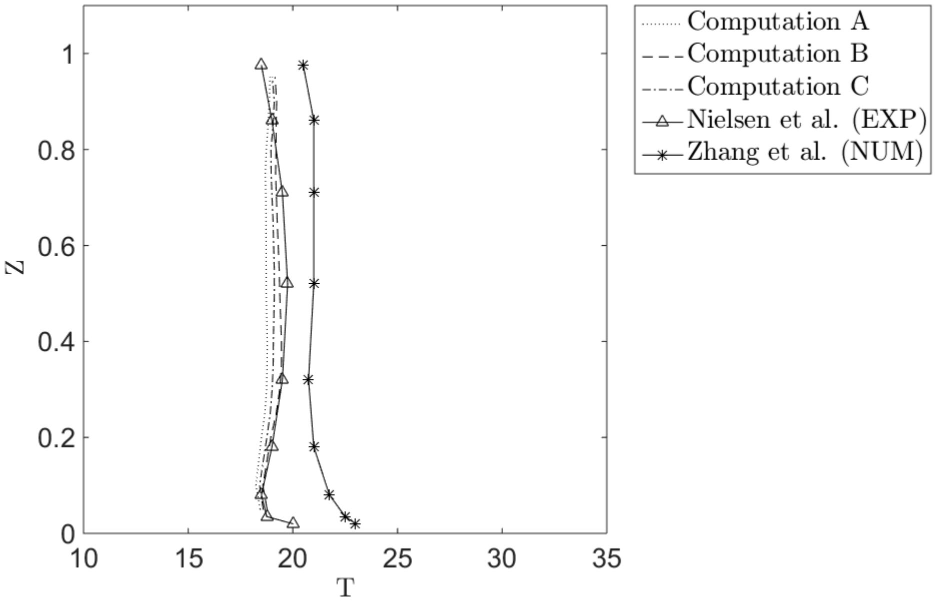


Figure 3.10 – Results on the temperature T at $x = 0.52\ m$, $y = 0.15\ m$ along the Z axis direction.

therefore the one proposed in *Computation B* that takes into account the Sutherland law and the turbulence sub-grid scale model to compute either the dynamic viscosity, and the thermal conductivity.

3.5.3 Conclusions on the validation case

To conclude, the 3D ventilated cavity benchmark case allows us to validate the numerical methods presented in this chapter, and used in Cimlib-CFD to solve aerothermal problems. In fact, the results presented here are in good agreement with experimental data. Furthermore, this benchmark study makes it possible for us to validate the implemented numerical models dedicated to the aerothermal coupling and to test three different types of coupling between the flow and its temperature. The coupling corresponding to *Computation B* was therefore selected to solve aerothermal problems in Cimlib-CFD and in particular, to solve the single impingement jet cooling case presented in the previous chapter.

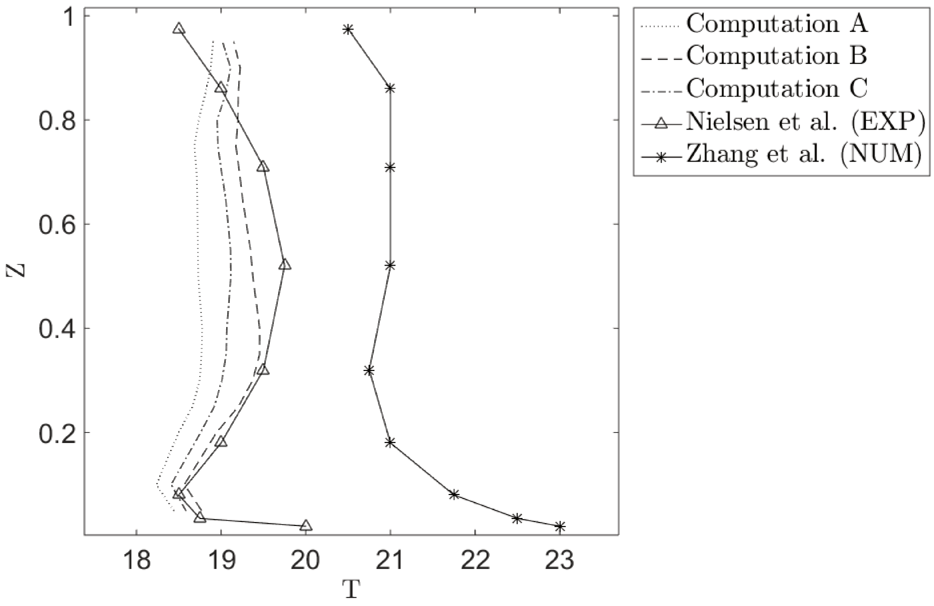


Figure 3.11 – Comparison between the three different aerothermal coupling solutions.

3.6 Aerothermal coupling applied to the impingement jet cooling

In this section, we propose to apply the additional numerical models dedicated to aerothermal coupling to the single impingement jet cooling. For this particular case, different types of aerothermal coupling models have been used in the literature. For example, the dynamic sub-grid scale model from Germano et al. has been used by Uddin et al. in [21]. However, Aillaud et al. in [24] propose to use another modified SM called the Wall Adapting Local Eddy-viscosity (WALE) model. In this work, we use the aerothermal coupling numerical models presented in Section 3.4 and validated in Section 3.5.

3.6.1 Results on the modified dynamic viscosity μ_T

A snapshot of the resulting dynamic viscosity μ_T is given in Figure 3.12. We observe the variations of μ_T along with the dynamic variations of the fluid temperature. As expected, the dynamic viscosity increases in the near wall region but the overall order of magnitude of its variations stays below 6% of the tabulated dynamic viscosity μ_{fluid}.

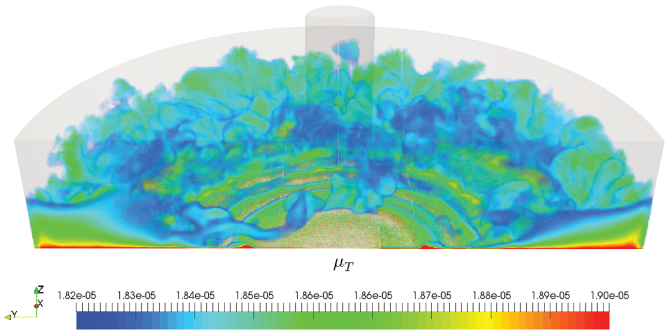


Figure 3.12 – Snapshot of the modified dynamic viscosity μ_T

3.6.2 Results on the modified thermal conductivity λ_{total}

The resulting modified thermal conductivity is presented in Figure 3.13. As expected, we observe that the thermal conductivity increases in the regions where important turbulent activity is expected. Furthermore, we can also observe the Van Driest damping function effects near the plate. Indeed, we observe that the modified thermal conductivity λ_{total} gets close to the tabulated thermal conductivity λ_{fluid} when we get close to the near plate area.

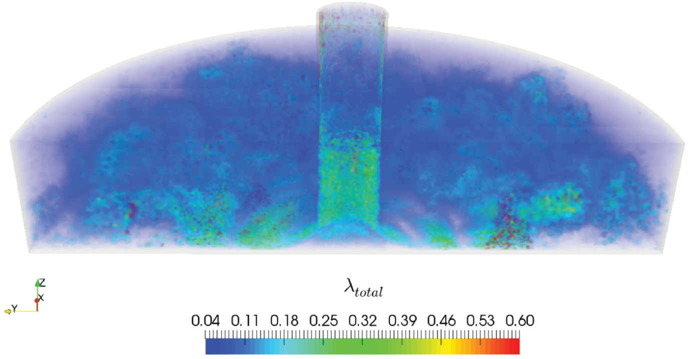


Figure 3.13 – Snapshot of the modified thermal conductivity λ_{total}

3.7 Conclusion

To conclude, we proposed in this chapter a set of numerical methods to solve aerothermal problems in Cimlib-CFD. These numerical methods made it possible for us to simulate different types of heat transfer in turbulent flows. In particular, we solved decoupled aerothermal problems between the velocity/pressure couple and the temperature. Therefore, additional numerical models were implemented to complete these numerical tools. We chose two additional numerical models which are (i) the Sutherland law and (ii) the Smagorinsky Model (SM).

A numerical study was then proposed to validate our numerical set of tools. The results were in good agreement with experimental and numerical references. In particular, they gave us the best way to couple the unsteady incompressible Navier-Stokes VMS solver with the thermal convection-diffusion solver using a modified dynamic viscosity μ_{total} and a modified thermal conductivity λ_{total}. Thus, we decide to integrate these additional numerical models to our simulations.

In the next chapter, we will propose an innovative adaptive multiscale method that allows dealing with highly convective aerothermal computations in Cimlib-CFD. Indeed, the next chapter will aim at proposing a new multiscale mesh adaptation technique that is well suited for the type of aerothermal problems developed in the first two chapters of this manuscript. This new method will be first applied to analytic benchmarks in 2D and 3D before being tested on a more realistic 3D industrial case.

Résumé du chapitre en français

Dans ce chapitre, nous introduisons les méthodes numériques développées dans Cimlib-CFD pour résoudre les problèmes aérothermiques turbulents. Nous résolvons deux systèmes aux dérivées partielles découplés. Le premier système est consacré à la résolution instationnaire et implicite des équations de Navier-Stokes incompressibles pour calculer les champs de vitesse et de pression. La résolution de ce système utilise une méthode éléments finis stabilisée venant de la formulation VMS.

Le second système concerne la résolution de l'équation thermique de convection-diffusion pour calculer la température du fluide. Cette résolution utilise la vitesse calculée dans le système d'équation précédent dans le terme de convection. Les transferts thermiques convectifs sont considérés dominants pendant la simulation. Pour cette raison, nous utilisons le schéma de résolution SUPG de manière à stabiliser la formulation éléments finis du problème thermique.

Nos systèmes d'équations étant découplés, nous proposons dans ce travail l'introduction de deux modèles numériques additionnels permettant de renforcer le couplage aérothermique de la simulation. Ces modèles sont: (i) la loi de Sutherland (i) et le modèle de sous-maille de Smagorinsky.

Ces méthodes numériques sont testées et validées sur un cas de cavité ventilée en 3D. Les résultats montrent qu'il y a une bonne corrélation entre nos résultats et les références numériques et expérimentales. En particulier, trois couplages aérothermiques différents sont testés dans le cas de la cavité 3D. Les résultats montrent que le meilleur couplage aérothermique est réalisé lorsque les deux modèles additionnels sont pris en compte dans le calcul de la viscosité dynamique et de la conductivité thermique du fluide.

Chapter 4

VMS Error Estimator for Anisotropic Adaptive Simulations

Contents

4.1 Introduction

The use of CFD for industrial applications has been in constant increase for the last decades. Researchers are continuously developing new techniques to reach higher level of precision. Nevertheless, to comply with industrial expectations, a trade-off has to be found between high precision levels and high computational costs [10]. Different strategies can be found in the literature. Most of them are related to high order elements (see [58–60]), parallel computing (see [35, 61–63]) or, in particular, adaptive methods (see [64–67]).

Indeed, adaptive methods make it possible to improve the accuracy and the efficiency of numerical methods. In particular, anisotropic mesh adaptation has proved to be powerful in capturing dynamically the heterogeneities that can appear in numerous physical applications including those having boundary or inner layers [68, 69]. In these cases, gradients of the solution are highly directional and can be captured with a good accuracy using fewer additional elements. These mesh adaptation techniques are based on local modifications of an existing mesh. Usually, it consists in a local stretching of the elements which is defined by a metric field. This metric field is built from an error analysis on the mesh. In [46, 70, 71] for example, the error analysis is done on the edges of the elements.

In fact, theories of anisotropic error estimation have been well developed, leading to some standardization of the adaptation process. Error estimation of the discretization error and in particular, of the interpolation error have been performed in a number of works such as in [72–74]. From the interpolation error analysis, several recent results [74–78] have brought renewed focus on metric-based mesh adaptation where the underling metric is derived from a recovered Hessian. Indeed, the Hessian based metric mesh adaptation has several advantages from which we note: (i) the general computation framework, (ii) the relatively easy way of implementation and above all, (iii) the robustness.

Despite the practical construction of this kind of anisotropic error estimators, the information derived from them is only an indicator for the mesh adaptation mechanics. Indeed, the fine scale features related to some dynamic flow solutions are still difficult to capture without considering the PDE-dependent approximation error. The estimation of the latter involves the development of multiscale a posteriori error estimators that can be computationally prohibitive.

However, the VMS method that we use to stabilize our continuous finite element scheme provides, by construction, a cost free PDE-dependent a posteriori error estimator. The VMS approach, introduced in the previous chapter, consists in the splitting of the solution into a resolved part (i.e. coarse scales) and an unresolved part (i.e. subscales). The resolution of the unresolved part gives a direct access to the sub-mesh scale information of the solution and allows us to compute an approximation error estimator without solving any additional equation. Recently, several works related to VMS (or subscales) error estimators in the VMS framework have been published. Granzow et al. in [79] developed an error representation for output quantities based on a dual enrichment technique. Also, in [80], Baiges et al. proposed a general error estimator for the finite element solution of solid mechanics problems. John et al. proposed in [81] a robust residual-based a posteriori estimator for the SUPG finite element method applied, in particular, to stationary convection-diffusion-reaction equations.

In this chapter, we compute the subscales error estimator using two different methods. The first one uses stabilizing parameters derived from the VMS analysis and local norms defined on the elements. It has been developed by Hauke et al. in [82, 83] and provides an *element-wise* computation of the error. Referring to Irisarri et al., the second method uses a linear combination of bubble functions to establish a *pointwise* computation of the error. It has been developed for one-dimensional transport equation in [84] and for 2D transport equation in [44].

The first contribution of this work is then to propose a new isotropic mesh adaptation technique based on the previous subscales error estimator. To do so, we define a new isotropic metric tensor $\mathscr{H}_{iso}$ and we solve an optimization problem under the constraint of a fixed number of elements. Until now, the VMS error estimators were only derived for isotropic mesh adaptation. In this work, we also propose to use the sub-mesh scale information for anisotropic mesh adaptation.

The second contribution is therefore to combine both the coarse scales interpolation error indicator and the subscales error estimator for anisotropic mesh adaptation. To the best of our knowledge, such strategy has never been tempted. To do so, we derive a new anisotropic metric tensor $\mathscr{H}_{aniso}^{new}$ that allows taking into account the anisotropic variations of the solution on the mesh and also relies upon the sub-grid information of the solution. This new anisotropic mesh adaptation technique takes into account the finite element VMS framework used for the numerical resolution of the convection-diffusion equation. In fact, the VMS approach allows quantifying the a posteriori subscales errors that can be used as weight to enrich the anisotropic error indicator of the conventional Hessian based approach. Unlike existing anisotropic a posteriori error estimators like in [85–87] to mention a few, this hybrid approach takes advantages from both a priori and a posteriori methods. Indeed, we build an isotropic a posteriori error estimator targeting the subscales and we combine it with an anisotropic a priori interpolation based error indicator targeting the coarse scales.

To begin this chapter, we refer to the VMS formulation applied to convection-diffusion problems introduced in Chapter 3, Section 3.3. From this mathematical background, we present, in Section 4.2, the two different computation methods of the subscales error estimator. Then, mesh adaptation is presented in Section 4.3. As a reminder, we introduce the Hessian based anisotropic mesh adaptation in Section 4.3.1. Then, we define the isotropic metric tensor $\mathscr{H}_{iso}$, built from the subscales error estimator in Section 4.3.2. Finally, we present the new combination of the interpolation error indicator and the subscales error estimator in Section 4.3.3. This combination is tested on 2D and 3D convection-diffusion benchmarks in Section 4.4. In this latter section, an error analysis of the exact error is made to highlight the two contributions of this chapter. Finally, we apply these new mesh adaptation techniques to the single impingement jet cooling in Section 4.5 in order to demonstrate the capabilities of these multiscale mesh adaptation techniques on a realistic industrial case.

4.2 A posteriori error estimation on solution's subscales

To begin, we can recall here that the VMS analysis introduced in Section 3.3.4 gives access to a solution of the subscales u'. Indeed, we gave an exact expression of u' in Eq. (3.35). Considering the relation $u = u_h + u'$, this part of the solution can also be considered as the *subscales error*. However, u' is not explicitly computed during the simulation. To obtain it, we use an a posteriori computation.

In this chapter, we use two different methods for the a posteriori computation of the subscales error estimator. Both methods are residual based and rely upon convection-dominated regime assumptions:

- The first method consists in computing the error estimator as the multiplication of the stabilizing parameter's norm and the residual's norm. The computation is done *element-wise* because of the local definition of the stabilizing parameter and the local error norms used.

- The second method does not use the stabilizing parameter. The error estimator is computed thanks to the *pointwise* error estimation of Irisarri et al. in [44]. It uses a set of bubble functions as a substitution of the subscales Green's functions.

In this section, we give the key features of these two computation methods.

4.2.1 Computation using stabilizing parameters

Going back to Section 3.3.4, we had the following sub-problem for the subscales:

$$a(w',u') = -(w',\mathscr{L}u_h - f) - (w',[\mathscr{B}u_h])_{\tilde{\Gamma}} - (w',\mathscr{B}u_h)_{\Gamma_h} \ \ \forall w' \in \mathscr{V}'$$

According to [45], this problem can be solved analytically using Green's function $g'(\mathbf{x},\mathbf{y})$. We have the following paradigm:

$$u'(\mathbf{x}) = -\int_{\tilde{\Omega}_y} g'(\mathbf{x},\mathbf{y})(\mathscr{L}u_h - f)(\mathbf{y})d\Omega_y - \int_{\tilde{\Gamma}_y} g'(\mathbf{x},\mathbf{y})([\mathscr{B}u_h])(\mathbf{y})d\Gamma_y - \int_{\Gamma_{hy}} g'(\mathbf{x},\mathbf{y})(\mathscr{B}u_h)(\mathbf{y})d\Gamma_y \quad (4.1)$$

This error estimator includes:

- Interior residuals, $\mathscr{L}u_h - f$ in $\tilde{\Omega}$
- Inter-elements residuals, $[\mathscr{B}u_h]$ on $\tilde{\Gamma}$
- Natural boundary condition residual $\mathscr{B}u_h$ on Γ_h

From here, multiple cases can be considered. In this work, we use linear shape functions with a piecewise constant source term f. Therefore, we can make the assumption that the residual $\mathscr{L}u_h - f$ is P0. Furthermore, we consider only Dirichlet boundary conditions. Thus, we neglect the effects of the boundary condition residual. Finally, as for all convection-dominated regimes, we will first consider the smooth case, i.e. that the discontinuity of the subscales is not preponderant and that the inter-elements residuals can be neglected. With these assumptions, we write:

$$u'(\mathbf{x}) \approx -\int_{\tilde{\Omega}_\mathbf{y}} g'(\mathbf{x},\mathbf{y})(\mathscr{L}u_h - f)(\mathbf{y})d\Omega_y \quad (4.2)$$

We will also make the assumption that the error has a local influence. Thus, we can replace the subscales Green's function by a local Green's function that vanishes at the element boundary, i.e., $g^e(\mathbf{x},\mathbf{y}) = 0$ on Γ^e. We can write:

$$g'(\mathbf{x},\mathbf{y}) \approx g^e(\mathbf{x},\mathbf{y}) \approx \tau_e \delta(\mathbf{y}-\mathbf{x}) \quad (4.3)$$

with,

$$\tau_e = \frac{1}{meas(\Omega_e)} \int_{\Omega_{e,\mathbf{x}}} \int_{\Omega_{e,\mathbf{y}}} g^e(\mathbf{x},\mathbf{y}) d\Omega_{e,\mathbf{x}} d\Omega_{e,\mathbf{y}} \quad (4.4)$$

In [47], the author shows that the above assumptions work well for convection-dominated regimes. In fact, numerical experiments show that for high Peclet number flows and stabilized solutions, the contribution of jump terms is negligible.

From here, we can therefore write a local error estimator like the following:

$$u'(\mathbf{x})|_{\Omega_e} \approx -\int_{\Omega_e} g^e(\mathbf{x},\mathbf{y})(\mathscr{L}u_h - f)(\mathbf{y})d\Omega_y = -\int_{\Omega_e} \tau_e \delta(\mathbf{y}-\mathbf{x})(\mathscr{L}u_h - f)(\mathbf{y})d\Omega_y = -\tau_e(\mathscr{L}u_h - f)(\mathbf{x}) \quad (4.5)$$

Taking the L^2 norm, we can finally write the following local error estimator for the solution subscales:

$$\boxed{||u'(\mathbf{x})||_{L^2,\Omega_e} = |\tau_e| \ ||\mathscr{L}u_h - f||_{L^2,\Omega_e}} \quad (4.6)$$

We know the expression of τ_e from the application of the SUPG method (see Section 3.3.3) and we can *a posteriori* compute the P0 residual in our finite element computation domain. As said in the introduction, the sub-scale information computed here is *element-wise* and can be used as such in mesh adaptation.

4.2.2 Computation using high order bubbles functions

In this section, we refer to the work of Irisarri et al. in [44]. This time, we apply a *pointwise* computation of the error estimator. This computation method consists in decomposing the error into two components according to the nature of the residuals:

$$u'(\mathbf{x}) = u'_{bub}(\mathbf{x}) + u'_{poll}(\mathbf{x}). \tag{4.7}$$

The first term u'_{bub} is the internal residual error and it is related to the local internal residual, $f - \mathscr{L} u_h$, inside the elements. As we will see later, this part of the error is modeled locally thanks to a set of bubble functions. The second term u'_{poll} is the inter-element error. It represents the pollution error due to sources of errors outside the element. As said in the previous section, it is negligible when considering convection-dominated regime. Consequently, in this work we will consider only the internal residual error and suppose that:

$$u'(\mathbf{x}) \approx u'_{bub}(\mathbf{x}). \tag{4.8}$$

4.2.2.1 Practical aspects

As proposed by Irisarri et al. in [44], the way of obtaining a numerical solution of u'_{bub} is by solving the following discrete problem:

$$\begin{cases} \text{Find } u'_{bub} \in \mathscr{S}^h_{bub} \text{ such that} \\ a(w'_{bub}, u'_{bub}) = (w'_{bub}, f - \mathscr{L} u_h), \quad \forall w'_{bub} \in \mathscr{S}^h_{bub} \end{cases} \tag{4.9}$$

This time, the error component is expressed with a combination of bubble functions b_i:

$$u'_{bub}(\mathbf{x}) = \sum_{i=1}^{n_{bub}} c_i^b b_i(\mathbf{x}) \tag{4.10}$$

Considering bubbles functions of order 3, we have:

$$u'_{bub}(\mathbf{x}) = c_1^b b_1(\mathbf{x}) + c_2^b b_2(\mathbf{x}) + c_3^b b_3(\mathbf{x}) \tag{4.11}$$

with c_i^b unknown constant to be determined.

Referring to [42], the definition of the first bubble function $b_1(\mathbf{x})$ is the following:

$$b_1(\mathbf{x}) = (d+1)^{d+1} \prod_{i=1}^{d+1} \hat{\lambda}_i \tag{4.12}$$

where d is the dimension of the problem and $\hat{\lambda}_i$ are the barycentric coordinates in the reference element.

The next bubble functions $b_2(\mathbf{x})$ and $b_3(\mathbf{x})$ are built by adding the monomials of the Pascal triangle with center in the barycenter $c_e = (\xi_e, \eta_e)$ of the element. For example, in 2D, in the reference element: $\Omega_{ref} = \{(\xi, \eta) : 0 \le \xi \le 1; 0 \le \eta \le 1 - \xi\}$, we choose the following bubble functions:

$$\begin{aligned} b_1(\xi, \eta) &= 27 \times \xi\eta(1-\xi-\eta) \\ b_2(\xi, \eta) &= 27 \times \xi\eta(1-\xi-\eta)(\xi - \xi_b) \\ b_3(\xi, \eta) &= 27 \times \xi\eta(1-\xi-\eta)(\eta - \eta_b) \end{aligned} \tag{4.13}$$

with $c_e = (\xi_b, \eta_b)$, $\xi_b = 1/3$ and $\eta_b = 1/3$.

Approximating $u'_{bub}(\mathbf{x})$ by Taylor series and neglecting the second order terms, we have an expression of $u'_{bub}(\mathbf{x})$ close to the centroid $\mathbf{c}_i$ of the element [44]:

$$u'_{bub}(\mathbf{x}) \approx b_1(\mathbf{x})(f - \mathscr{L}u_h)(\mathbf{c}_i) + b^e_{y_1}(\mathbf{x})\frac{d(f-\mathscr{L}u_h)}{dy_1}|_{y=c_i} + b^e_{y_2}(\mathbf{x})\frac{d(f-\mathscr{L}u_h)}{dy_2}|_{y=c_i} + \mathrm{HO} \tag{4.14}$$

where:

$$b^e_{y_1} = \int_{\Omega_e} g^e(\mathbf{x},\mathbf{y})(y_1 - c_{i,1})d\Omega_y \quad and \quad b^e_{y_2} = \int_{\Omega_e} g^e(\mathbf{x},\mathbf{y})(y_2 - c_{i,2})d\Omega_y \tag{4.15}$$

As said before, we make the assumption that the residual $f - \mathscr{L}u_h$ is P0. Therefore, we have:

$$\frac{d(f-\mathscr{L}u_h)}{dy_1}|_{y_1=c_i} = \frac{d(f-\mathscr{L}u_h)}{dy_2}|_{y_2=c_i} = 0 \tag{4.16}$$

Consequently, the internal residual can be simply expressed as follows:

$$u'_{bub}(\mathbf{x}) \quad = b_1(\mathbf{x})(f - \mathscr{L}u_h)(\mathbf{c}_i) \tag{4.17}$$

Developing the residual with the convection-diffusion generic operator $\mathscr{L}$, we have:

$$u'_{bub}(\mathbf{x}) \quad = b_1(\mathbf{x})(f(\mathbf{c}_i) + a\Delta u_h(\mathbf{c}_i) - \mathbf{v}\cdot\nabla u_h(\mathbf{c}_i)) \tag{4.18}$$

Then, u_h is P1, therefore, $\nabla u_h(\mathbf{c}_i)$ is a constant inside the element and $\Delta u_h(\mathbf{c}_i) = 0$. For this reason, we finally get:

$$u'_{bub}(\mathbf{x}) = b_1(\mathbf{x})(f(\mathbf{c}_i) - \mathbf{v}\cdot\nabla u_h(\mathbf{c}_i)) \tag{4.19}$$

The above computation of the error estimator is *pointwise*. In fact, the error estimator is given at each point $\mathbf{x}$ of the domain. However, to include the error information in the mesh adaptation, we need an information of the error inside each element. To get this information, we can compute 3 different types of norm: L^1, L^2 and L^∞. We recall here the definition of these 3 norms:

$$||u'_{bub}||_{\mathrm{L}^1}(\Omega_e) = |\Omega_e| \times \sum_{1\leqslant i\leqslant \mathrm{N}_{interp}} |u'_{bub}(\mathbf{x}_i)| \tag{4.20}$$

$$||u'_{bub}||_{\mathrm{L}^2}(\Omega_e) = \sqrt{|\Omega_e|} \times \sqrt{\sum_{1\leqslant i\leqslant \mathrm{N}_{interp}} |u'_{bub}(\mathbf{x}_i)|^2} \tag{4.21}$$

$$||u'_{bub}||_{\mathrm{L}^\infty}(\Omega_e) = max\left\{ |u'_{bub}(\mathbf{x})| \;\mid\; \mathbf{x} = (\mathbf{x}_i)_{1\leqslant i\leqslant \mathrm{N}_{interp}}\right\} \tag{4.22}$$

where $(\mathbf{x}_i)_{1\leqslant i\leqslant \mathrm{N}_{interp}}$ are the interpolation points defined by the integration points of the bubble function $b_1(\mathbf{x})$ and N_{interp} is the number of interpolation points. The consistency of this error estimate resides in the fact that it goes to zero when the interior residual goes to zero.

4.2.2.2 Validation of computation using bubbles functions

Now that we have developed practical computation aspects, we validate the computation on a benchmark of a convection-diffusion equation. To do so, we take the example given in [44] of an error estimation where the FEM is stabilized by the SUPG scheme. We recall that the differential operator is given by: $\mathscr{L}u = -a\Delta u + \mathbf{v}\cdot\nabla u$ with $a = 0.03$, $f = 1$, $\mathbf{v} = (1,1)$ and homogeneous Dirichlet boundary conditions.

We compute the SUPG solution on P1 triangular elements instead of the Q2 bilinear quadrilateral elements used in the referred paper. Therefore, we choose a number of elements in order to get approximately the same SUPG solution in comparison with Irisarri et al.. We choose to divide each square cell in 8 triangles and we obtain a mesh of 512 triangular elements. We compare our

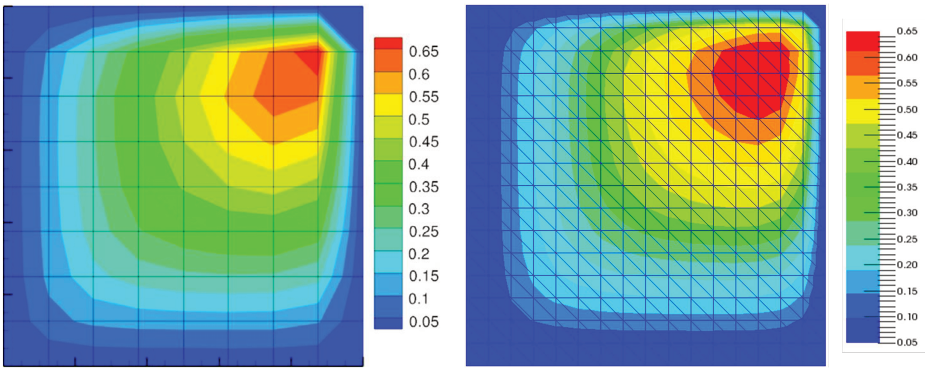


Figure 4.1 – Comparison of the SUPG solution between Irisarri et al. [44] (left) and our solution (right)

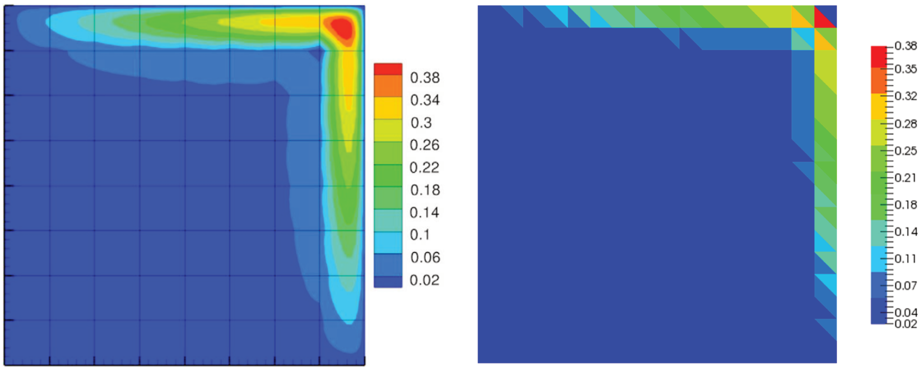


Figure 4.2 – Comparison of error estimate between Irisarri et al. [44] (left) and our error estimation (right)

SUPG solution with the reference in Figure 4.1.

Then, we compute the error estimate $||u'_{bub}||_{\mathrm{L}^\infty}(\Omega_e)$ as explained in the previous subsection. We compare our results with the referred paper in Figure 4.2.

Remark 3 . *As the reader will notice, we don't have the same graphic representation between the two figures. In fact, the graphic representation of Irisarri et al. keep the pointwise definition of the error estimator. It is therefore a point cloud representation. On the contrary, we use an element-wise graphic representation. We use this representation because, as we will see, we only need the maximum value inside the element for our use of the error estimator in mesh adaptation.*

Despite the difference in graphic representations, we observe a good concordance between the computation of Irisarri et al. and our computation. In fact, we reach approximately the same maximal values inside the elements and the error is located in the same area for both computations. Therefore, we validate the computation of the error estimator with bubble functions. We will see in the next section how to use this information in mesh adaptation.

4.3 Mesh adaptation with the subscales error estimator

In this section, we propose 3 different types of mesh adaptation:

1. An isotropic mesh adaptation: This technique uses the previously described subscales error estimator. From the latter, we build an isotropic metric tensor $\mathscr{H}_{iso}$ to adapt the mesh.

2. An anisotropic mesh adaptation (described in [35]): This technique uses the anisotropic local interpolation error indicator. From the latter, we derived an anisotropic metric tensor $\mathcal{H}_{aniso}$ to adapt the mesh.

3. A *new* anisotropic mesh adaptation: This technique uses a *new* anisotropic local error indicator that takes into account (i) the interpolation error indicator and (ii) the subscales error estimator. From the latter, we derive a new metric tensor $\mathcal{H}^{new}_{aniso}$.

As a reminder, we start by introducing the principles of anisotropic mesh adaptation and the construction of the anisotropic metric tensor $\mathcal{H}_{aniso}$. Then, the isotropic tensor $\mathcal{H}_{iso}$ based on the subscales error estimator is defined in Section 4.3.2. Finally, a combination between the anisotropic local error indicator and the subscales error estimator is proposed and the new metric tensor $\mathcal{H}^{new}_{aniso}$ is defined in Section 4.3.3.

4.3.1 Principles of anisotropic mesh adaptation

To discretize our computational domain, we use anisotropic unstructured meshes. It implies that we can stretch the elements in certain directions according to the solution features. To do so, we start by performing an error analysis on the mesh. Then, to correlate the error with the geometry, a metric field is defined. From this metric field, an anisotropic error indicator is defined and used as a functional for a re-meshing optimization problem.

Let us consider a certain triangulation Ω_h. We can derive an upper bound of the approximation error using an interpolation error analysis in the L^p norm. Referring to Almeida et al. in [72], this upper bound is expressed thanks to the recovered Hessian of the approximated solution u_h. In fact, using P1 linear elements, we usually cannot compute directly the Hessian of the solution. Instead, we compute an approximation called the *recovered* Hessian matrix:

$$||u - u_h||_{\mathrm{L}^p(\Omega)} \leq \mathrm{C}'\mathcal{N}_{\Omega_h}^{-\alpha}||\mathrm{H_R}(u_h)(x)||^p_{\mathrm{L}^p(\Omega_e)} \tag{4.23}$$

where $\alpha \geq 0$, $\mathcal{N}_{\Omega_h}$ the number of elements of the mesh for the triangulation Ω_h, $\mathrm{H_R}(u_h)(x)$ the *recovered* Hessian matrix and C' is independent of the element size.

To apply the re-meshing strategy, we build an equilateral tetrahedron in the metric space. As in [35], it is defined at an arbitrary point P by the local metric field $\mathcal{M}$:

$$\mathcal{M}(\mathrm{P}) = \frac{1}{h_1(\mathrm{P})} e_1 \otimes e_1 + ... + \frac{1}{h_d(\mathrm{P})} e_d \otimes e_d \tag{4.24}$$

with $(e_i)_{i=1,d}$ the eigenvectors of $\mathrm{H_R}(u_h(x))$ and $h_i(\mathrm{P})$ the mesh sizes in the e_i directions.

However, the *recovered* Hessian matrix is not a metric because it is not positive definite. Therefore, we define the following metric tensor:

$$\mathcal{H}_{aniso} = \mathcal{R}\Lambda\mathcal{R}^{\mathrm{T}} \tag{4.25}$$

where $\mathcal{R}$ is the orthogonal matrix built with the eigenvectors $(e_i)_{i=1,d}$ of $\mathrm{H_R}(u_h(x))$ and Λ is the diagonal matrix of absolute value of the eigenvalues λ_i of $\mathrm{H_R}(u_h(x))$. This metric tensor can also be written as follow:

$$\mathcal{H}_{aniso} = \mathcal{R}\Lambda\mathcal{R}^{\mathrm{T}} = |\lambda_1| e_1 \otimes e_1 + ... + |\lambda_d| e_d \otimes e_d \tag{4.26}$$

Here, we want to align the mesh with the solution field u. It means that we want the error to be equi-distributed on each direction of the domain. To do so, the shape of each element have to be such that the local error is equal in any direction. It is equivalent to say that the local error is constant per element in the principal directions of curvature. Therefore, we have:

$$|\lambda_1| h_1^2 = ... = |\lambda_d| h_d^2 = cte \tag{4.27}$$

Following the work of Mesri et al. in [35], we introduce the following local error indicator η_{Ω_e} in the L^p norm:

$$\eta_{\Omega_e} = d|\Omega_e|^{\frac{1}{p}}|\lambda_d(x_0)|h_d^2 \tag{4.28}$$

where $|\lambda_d(x_0)|$ is the maximum eigenvalues of $H_R(u_h(x))$ corresponding to direction d, $|\Omega_e|$ is the volume of the element and h_d is the length of the element in direction d.

In [35], the authors define a minimization problem where the functional is the error indicator of Eq. (4.28). This optimization problem is expressed as follows:

$$\left\{\begin{array}{l}\text{Find } h_{\Omega_e} = \{h_{1,\Omega_e},\dots,h_{d,\Omega_e}\},\ \Omega_e \in \Omega_h \text{ that minimizes the cost function:}\\ F(h_{\Omega_e}) = \sum\limits_{\Omega_e\in\Omega_h} (\eta_{\Omega_e})^p\\ \text{under the constraint:}\\ \mathcal{N}_{\Omega'_h} = C_0^{-1}\sum\limits_{\Omega_e\in\Omega_h}\int_{\Omega_e}\prod\limits_{i=1}^{d}\frac{1}{h_{i,\Omega_e}}d\Omega_e\\ \text{where } C_0 \text{ is the volume of a regular tetrahedron and } \Omega'_h \text{ is the new triangulation.}\end{array}\right. \tag{4.29}$$

Then, this multidimensional optimization problem is replaced by a one-dimensional optimization problem. The unknown is no more h_{Ω_e} but the mesh size corresponding to the max of the $(h_{i,\Omega_e})_{1\leqslant i\leqslant d}$. The way of dealing with anisotropy is via stretching factors $(s_{i,\Omega_e})_{1\leqslant i\leqslant d-1}$ defined between the mesh sizes $(h_{i,\Omega_e})_{1\leqslant i\leqslant d}$. The solution is given by Mesri et al. in [35]. In 3D and L^p norm, it is expressed thanks to the following theorem:

Theorem 1 *For $d = 3$, the optimization problem (4.29) has a unique solution and is given by:*

$$\left\{\begin{array}{l} h_{3,\Omega_e} = \left[\dfrac{\beta}{\frac{(2p+3)}{3}C_{1,\Omega_e}}\displaystyle\int_{\Omega_e} C_{2,\Omega_e}d\Omega_e\right]^{\frac{1}{2(p+3)}}\\ h_{2,\Omega_e} = s_{2,\Omega_e}h_{3,\Omega_e}\\ h_{1,\Omega_e} = s_{1,\Omega_e}s_{2,\Omega_e}h_{3,\Omega_e}\end{array}\right.$$

with

$$\left\{\begin{array}{l} s_{1,\Omega_e} = \dfrac{h_1}{h_2} = \left(\dfrac{|\lambda_2|}{|\lambda_1|}\right)^{\frac{1}{2}}\\ s_{2,\Omega_e} = \dfrac{h_2}{h_3} = \left(\dfrac{|\lambda_3|}{|\lambda_2|}\right)^{\frac{1}{2}}\end{array}\right.$$

where

$$C_{1,\Omega_e} = 3^pC_0s_{1,\Omega_e}s_{2,\Omega_e}^2|\lambda_3|^p,\quad C_{2,\Omega_e} = C_0^{-1}\frac{1}{s_{1,\Omega_e}s_{2,\Omega_e}^2}\quad \textit{and}$$

$$\beta^{\frac{1}{\frac{2}{3}(p+3)}} = \mathcal{N}_{\Omega'_h}^{-1}\sum_{\Omega_e\in\Omega_h}\left\{\left(\frac{1}{\int_{\Omega_e}C_{2,\Omega_e}d\Omega_e}\right)^{\frac{1}{\frac{2}{3}(p+3)}}\int_{\Omega_e}C_{2,\Omega_e}\left[\frac{2p+3}{3}C_{1,\Omega_e}\right]^{\frac{1}{\frac{2}{3}(p+3)}}d\Omega_e\right\}$$

Finally, the above solution gives the mesh sizes in the 3 directions that define the metric field on each element. These mesh sizes are computed with respect to a fixed number of elements $\mathcal{N}_{\Omega'_h}$.

4.3.2 Isotropic mesh adaptation with the subscales error estimator

The first use of the subscales error estimator computed in Section 4.2 is for an *isotropic* mesh adaptation. Referring to [88] and from a theoretical convergence point of view, we know that the error is linked to the local mesh size h according to the following relation:

$$||u'||_{L^\infty(\Omega)} \approx C.h^{k+1} \tag{4.30}$$

where C is independent of the element size. In our case, using linear elements, we have:

$$||u'||_{\mathrm{L}^\infty}(\Omega) \approx \mathrm{C}.h^2 \tag{4.31}$$

To control the mesh size of our adaptive process, the user has to give a tolerance value $||u'_{tol}||_{\mathrm{L}^\infty}$ so that the local mesh size stays above an acceptable bound. In fact, this tolerance corresponds to the desired error that the user wants to obtain on the mesh. For example, in a *pointwise* error estimation, $||u'_{tol}||_{\mathrm{L}^\infty}$ is the tolerated error at each control point inside the element. Referring to [89], we can write the tolerated pointwise error as a scalar positive value u'_{TOL}:

$$||u'_{tol}||_{\mathrm{L}^\infty} = u'_{\mathrm{TOL}} \tag{4.32}$$

Starting from here, we can consider three strategies to adapt the mesh. Indeed, we can decide to either:

1. uniformly distribute the desired local error norm over the *old* mesh or,
2. uniformly distribute the desired local error norm over the *new* mesh or,
3. uniformly distribute the desired pointwise error.

In this work, we will use the 2^{nd} mesh adaptation strategy for the *element-wise* computation of the error estimator (i.e. the one using stabilizing parameters); and the 3^{rd} mesh adaptation strategy for the *pointwise* computation of the error estimator (i.e. the one using bubble functions).

For the 3^{rd} strategy, still referring to [89], we write the relationship between the error and the tolerance in a uniform pointwise error distribution:

$$\left(\frac{h_{new}}{h}\right)^2 = \frac{u'_{\mathrm{TOL}}}{u'} \tag{4.33}$$

where h_{new} is the size of the new mesh (after mesh adaptation), u' is the estimated error.

Then, we build the isotropic metric tensor:

$$\mathscr{H}_{iso} = \mathscr{R}\Lambda\mathscr{R}^{\mathrm{T}} = |\lambda| e_1 \otimes e_1 + ... + |\lambda| e_d \otimes e_d \tag{4.34}$$

with

$$|\lambda| = \frac{1}{h_{new}^2} = \frac{||u'||_{\mathrm{L}^\infty}(\Omega_e)}{u'_{\mathrm{TOL}}} \times \frac{1}{h^2} \tag{4.35}$$

where $\mathscr{R}$ is the orthogonal matrix built with the eigenvectors $(e_i)_{i=1,d}$ of $\mathrm{H_R}(u_h(x))$.

Here, the eigenvalues λ of the metric are equal in all directions. Therefore, the mesh will be adapted *isotropically*. However, we keep the eigenvectors of $\mathrm{H_R}(u_h(x))$ in the definition of $\mathscr{H}_{iso}$. Thus, we keep the element's orientation prescribed by the *recovered* Hessian matrix. Finally, the re-meshing strategy is the same that the one described in Section 4.3.1. In particular, as we apply an isotropic re-meshing, the stretching factors $(s_{i,\Omega_e})_{1\leqslant i\leqslant d-1}$ between the mesh sizes $(h_{i,\Omega_e})_{1\leqslant i\leqslant d}$ are equal to 1.

4.3.3 Combination of subscales error estimator with anisotropic mesh adaptation

The second use of the subscales error estimator computed in Section 4.2 is for an *anisotropic* mesh adaptation. In this section, we propose to combine both the coarse scales error indicator and the subscales error estimator. To do so, we derive a new metric that allows taking into account the anisotropic variations of the solution on the mesh but also relies on the subscales error estimator previously computed.

We first consider the previous anisotropic local error indicator η_{Ω_e} defined in Eq. (4.28) and recalled here:

$$\eta_{\Omega_e} = d|\Omega_e|^{\frac{1}{p}}|\lambda_d(x_0)|h_d^2$$

Then, going back to the previous section, we write:

$$\frac{h_{d,new}^2}{h_d^2} = \frac{u'_{\mathrm{TOL}}}{||u'||_{\mathrm{L}^\infty(\Omega_e)}}$$

Now, the unknown of the re-meshing problem is $h_{d,new}$. In fact, we want the new mesh size to take into account the error estimator of the subscales. Thus, we propose a new anisotropic local error indicator $\eta_{\Omega_e,new}$ defined as:

$$\eta_{\Omega_e,new} = d|\Omega_e|^{\frac{1}{p}} \times |\lambda_d(x_0)| \times h_{d,new}^2 \tag{4.36}$$

From here, we can define the new anisotropic metric tensor as:

$$\mathscr{H}_{aniso}^{new} = \mathscr{R}\Lambda\mathscr{R}^{\mathrm{T}} = \frac{||u'||_{\mathrm{L}^\infty(\Omega_e)}}{u'_{\mathrm{TOL}}}|\lambda_1|e_1 \otimes e_1 + \ldots + \frac{||u'||_{\mathrm{L}^\infty(\Omega_e)}}{u'_{\mathrm{TOL}}}|\lambda_d|e_d \otimes e_d \tag{4.37}$$

Doing so, we keep the anisotropic effects from the solution variations but we isotropically scale this effect by the subscales error estimator.

With this new error indicator, we solve the same optimization problem of Eq. 4.29 with the unknown $h_{d,new}$. The optimal mesh is obtained exactly in the same way but using $\eta_{\Omega_e,new}$ as functional. Consequently, the optimization problem becomes:

$$\left\{\begin{array}{l}\text{Find } h_{\Omega_e,new} = \{h_{1,\Omega_e,new},\ldots,h_{d,\Omega_e,new}\},\ \Omega_e \in \Omega_h \text{ that minimizes the cost function:}\\[1ex] \mathrm{F}(h_{\Omega_e,new}) = \displaystyle\sum_{\Omega_e\in\Omega_h}(\eta_{\Omega_e,new})^p\\ \text{under the constraint:}\\[1ex] \mathscr{N}_{\Omega'_h} = \mathrm{C}_0^{-1}\displaystyle\sum_{\Omega_e\in\Omega_h}\int_{\Omega_e}\prod_{i=1}^{d}\frac{1}{h_{i,\Omega_e,new}}d\Omega_e\\ \text{where } \mathrm{C}_0 \text{ is the volume of a regular tetrahedron and } \Omega'_h \text{ is the new triangulation.}\end{array}\right. \tag{4.38}$$

Following the same proof that Mesri et al. in [35], the solution of this new optimization problem becomes:

Theorem 2 *For $d = 3$, the optimization problem (4.38) has a unique solution and it is given by:*

$$\left\{\begin{array}{l} h_{3,\Omega_e,new} = \left[\dfrac{\beta}{\frac{(2p+3)}{3}\mathrm{C}_{1,\Omega_e,new}}\displaystyle\int_{\Omega_e}\mathrm{C}_{2,\Omega_e}d\Omega_e\right]^{\frac{1}{2(p+3)}}\\ h_{2,\Omega_e,new} = s_{2,\Omega_e}h_{3,\Omega_e,new}\\[1ex] h_{1,\Omega_e,new} = s_{1,\Omega_e}s_{2,\Omega_e}h_{3,\Omega_e,new}\end{array}\right.$$

with

$$\left\{\begin{array}{l} s_{1,\Omega_e} = \dfrac{h_1}{h_2} = \left(\dfrac{|\lambda_2|}{|\lambda_1|}\right)^{\frac{1}{2}}\\[2ex] s_{2,\Omega_e} = \dfrac{h_2}{h_3} = \left(\dfrac{|\lambda_3|}{|\lambda_2|}\right)^{\frac{1}{2}}\end{array}\right.$$

where

$$\mathrm{C}_{1,\Omega_e,new} = 3^p\mathrm{C}_0 s_{1,\Omega_e}s_{2,\Omega_e}^2 \times \left(\frac{||u'||_{\mathrm{L}^\infty(\Omega_e)}}{u'_{\mathrm{TOL}}}\right)^p \times |\lambda_3|^p,\ \ \mathrm{C}_{2,\Omega_e} = \mathrm{C}_0^{-1}\frac{1}{s_{1,\Omega_e}s_{2,\Omega_e}^2} \quad \textit{and}$$

$$\beta^{\frac{1}{\frac{2}{3}(p+3)}} = \mathscr{N}_{\Omega'_h}^{-1}\sum_{\Omega_e\in\Omega_h}\left\{\left(\frac{1}{\int_{\Omega_e}\mathrm{C}_{2,\Omega_e}d\Omega_e}\right)^{\frac{1}{\frac{2}{3}(p+3)}}\int_{\Omega_e}\mathrm{C}_{2,\Omega_e}\left[\frac{2p+3}{3}\mathrm{C}_{1,\Omega_e,new}\right]^{\frac{1}{\frac{2}{3}(p+3)}}d\Omega_e\right\}$$

By re-defining the optimization problem, we integrate the effect of the subscales error estimator with respect to the constraint on the fixed number of elements. This new re-meshing strategy is tested on 2D and 3D benchmarks in the next section.

4.4 Numerical examples

In this section, we test the effectivity and the accuracy of our new local error indicator on 2D and 3D convection-diffusion benchmark cases in a convection-dominated regime. To do so, we start by computing the FE solution u_h thanks to the SUPG scheme described in this thesis. The solution is computed on 6 different types of meshes:

- ***Fixed mesh:*** Using no mesh adaptation,
- ***Isotropic mesh 1:*** Using the isotropic metric tensor $\mathcal{H}_{iso}$ computed with the subscales error estimator using the stabilizing parameter (see Section 4.2.1),
- ***Isotropic mesh 2:*** Using the isotropic metric tensor $\mathcal{H}_{iso}$ computed with the subscales error estimator using the bubble functions (see Section 4.2.2),
- ***Anisotropic mesh 1:*** Using the anisotropic metric tensor $\mathcal{H}_{aniso}$ computed *without* the subscales error estimator,
- ***Anisotropic mesh 2:*** Using the new anisotropic metric tensor $\mathcal{H}_{aniso}^{new}$ computed with the subscales error estimator using the stabilizing parameter,
- ***Anisotropic mesh 3:*** Using the new anisotropic metric tensor $\mathcal{H}_{aniso}^{new}$ computed with the subscales error estimator using the bubble functions.

For the benchmark cases 4.4.1 and 4.4.3, we have a known analytic solution u. Thus, we can compute the exact discrete error $e_h = u - u_h$ on each element of the mesh. With this local discrete error, we compute the global L^2 norm of the error such that:

$$||e_h||_{\mathrm{L}^2} = \left(\int_{\Omega_h} e_h^2 \, d\Omega_h \right)^{1/2} \tag{4.39}$$

For the case 4.4.1, we define an efficiency index I_{eff} for our subscales error estimator as follow:

$$\mathrm{I}_{eff} = \frac{||u'||_{\mathrm{L}^2}}{||e_h||_{\mathrm{L}^2}} \tag{4.40}$$

4.4.1 Case 1: regular boundary layers in 2D

This first case has a continuous solution and regular boundary layers. It has been studied by several authors like Zhang et al. in [90] or Hachem et al. in [46]. We consider the domain $\Omega = (0,1)^2$ and the velocity field $\mathbf{v}(x,y) = (1,1)^{\mathrm{T}}$. Instead of a source term $f = 1$ like in Section 4.2.2.2, we choose the source term corresponding to the following analytic solution:

$$u(x,y) = xy\left(1 - e^{-\frac{1-x}{a}}\right)\left(1 - e^{-\frac{1-y}{a}}\right) \tag{4.41}$$

Thus, we have the following source term:

$$f(x,y) = (x+y)\left(1 - e^{-\frac{1-x}{a}}\, e^{-\frac{1-y}{a}}\right) + (x-y)\left(e^{-\frac{1-y}{a}} - e^{-\frac{1-x}{a}}\right) \tag{4.42}$$

We compute the FE SUPG solution u_h for two different diffusion coefficients: $a = \{10^{-3}; 10^{-4}\}$. In addition, for practical purpose, we define the following stretching factor S_{Ω_e}:

$$\mathrm{S}_{\Omega_e} = \frac{h_{new}^2}{h^2} = \frac{u'_{\mathrm{TOL}}}{||u'||_{\mathrm{L}^\infty(\Omega_e)}} \tag{4.43}$$

To keep an acceptable mesh even where $||u'||_{L^\infty}(\Omega_e)$ is close to zero, we have to impose a lower bound for the stretching factor S_{Ω_e}. We decide that, from an iteration to the other, the size of the element should not increase of more than 10 times the previous one. Therefore, the condition: $S_{\Omega_e} \geqslant 0.01$ is applied.

The effectivity index for the two ways of computation of the error estimator is given in Figure 4.3.

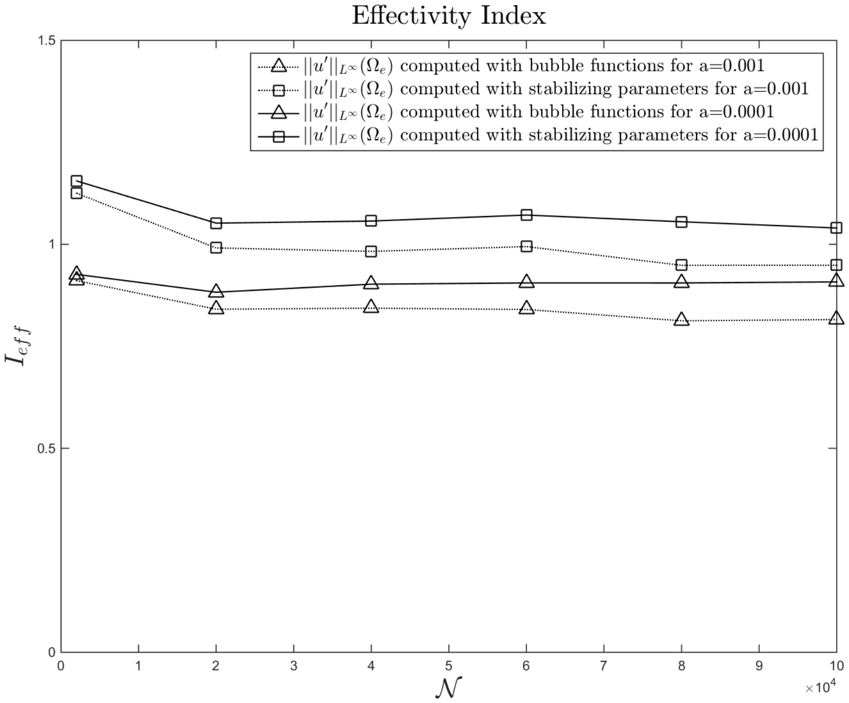


Figure 4.3 – Effectivity indexes for $a = 10^{-3}$ and $a = 10^{-4}$

The first observation that can be made concerning the efficiency index I_{eff} is that for the two different Peclet numbers and for the different mesh sizes considered, our efficiency index I_{eff} stays close to 1. Therefore, we can say that our error estimator is effective for all configurations. Secondly, we observe that the error estimation computed with bubble functions is more effective than the one computed with stabilizing parameters which overestimates the error. Finally, we observe that the effectivity index I_{eff} gets closer to 1 when the diffusion coefficient is reduced (i.e. when the element Peclet number increases). This finding is in accordance with the convection-dominated regime assumption.

Now that we have confirmed the effectivity of our error estimation, we can use this estimation in the mesh adaptation. To analyze the mesh convergence of u_h, we compute u_h for 6 different mesh sizes: 2 000, 20 000, 40 000, 60 000, 80 000 and 100 000 elements.

The distributions of the discrete error e_h are presented in Figure 4.4. The corresponding meshes for each error distribution are given below. On this figure, the case corresponds to a diffusion coefficient of $a = 10^{-3}$. Mesh adaptation is under a constraint of 20 000 elements and the figures are respectively:

- Figure 4.4.(a) corresponding to the *Fixed mesh*,
- Figure 4.4.(b) corresponding to the *Isotropic mesh 1*,
- Figure 4.4.(c) corresponding to the *Isotropic mesh 2*,

- Figure 4.4.(d) corresponding to the *Anisotropic mesh 1*,
- Figure 4.4.(e) corresponding to the *Anisotropic mesh 2*,
- Figure 4.4.(f) corresponding to the *Anisotropic mesh 3*.

To begin the analysis of the results on the error distributions, we compare the Figure 4.4.(a), (b) and (c). We notice that the *isotropic* mesh adaptation according to our subscales error estimator reduces the error in the boundary layer. In fact, we notice from Figure 4.4.(b) and (c) that the mesh is isotropically adapted in the boundary layer. However, we observe that the error stays above the prescribed tolerance $u'_{\mathrm{TOL}} = 0.01$ when we get close to the wall. To counter this effect, *anisotropic* mesh adaptation is a really efficient technique. In Figure 4.4.(d), we observe that the anisotropic mesh adaptation makes it possible for the error to go under the tolerance. However, we still keep an error approximately equals to the tolerance on the right top angle.

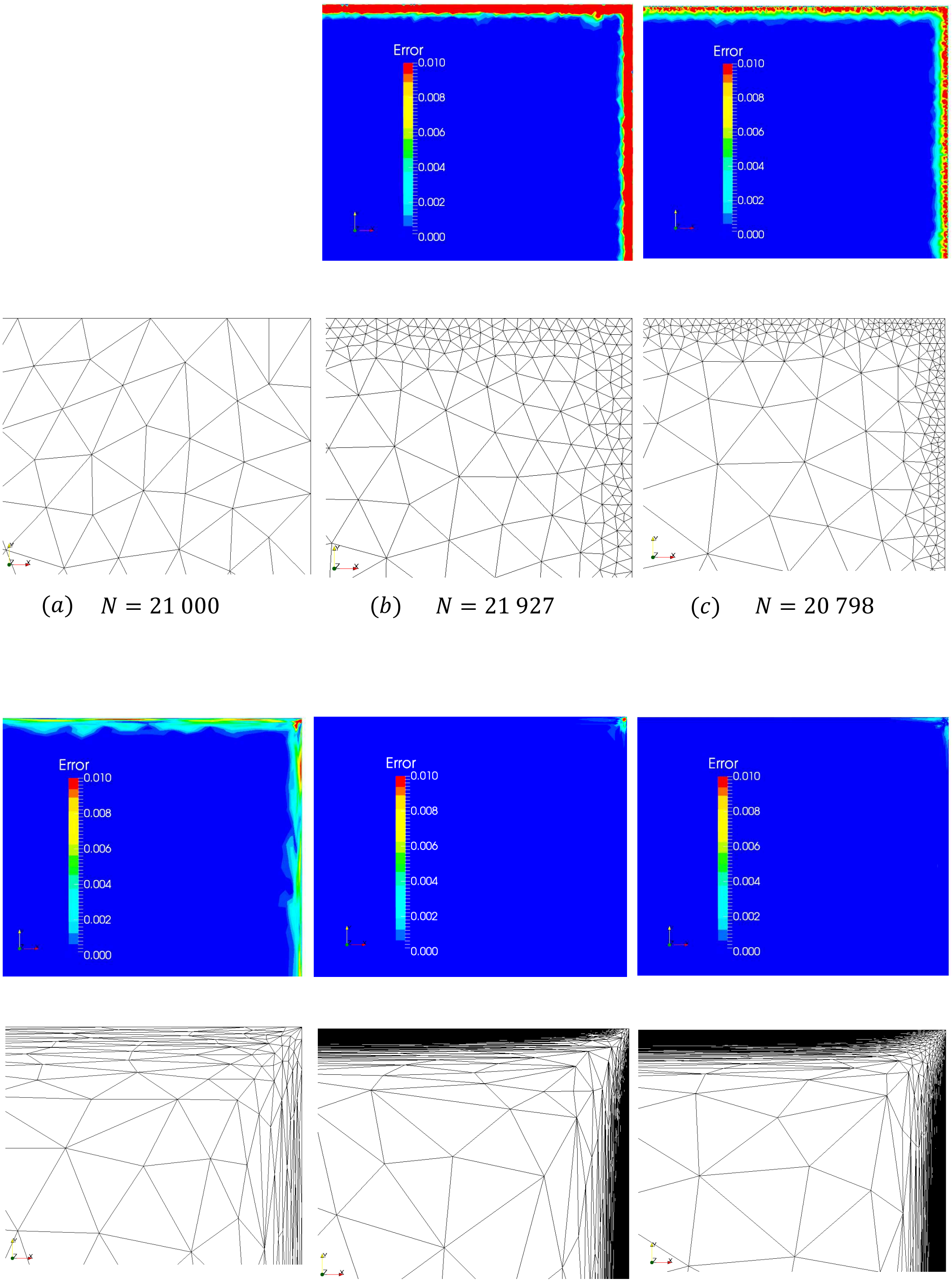
To eliminate this last part of the error, the new local error estimator proposed in Section 4.3.3 is well adapted as one can see in Figure 4.4.(f). In fact, we notice from the comparison of these 6 figures that our new local anisotropic error indicator $\eta_{\Omega_e,new}$ is the best driver of mesh adaptation. We observe in Figure 4.4.(e) and 4.4.(f) a drastic reduction of the error in the boundary layers compared to the other mesh adaptation techniques. Therefore, the use of the sub-scale information into the new anisotropic error indicator allows improving the locality of the mesh nodes with respect to the equi-distribution of different error scales. Furthermore, we see that the equi-distribution is optimal when the error estimator is computed with the bubble functions (see Figure 4.4.(f)). To finish, we notice that the constraint of a fixed number of elements is well respected with a deviation of less than 19% with the targeted number of elements $\mathcal{N}$.

The reduction of the error can also be observed on the results for the global L^2 norm of the error $||e_h||_{\mathrm{L}^2}$. This is true for the case $a = 10^{-3}$ in Figure 4.5 and the case $a = 10^{-4}$ in Figure 4.6. As expected, the *isotropic* mesh adaptation reduces the error for both cases of a. The error estimator computed with bubble functions shows to be the best driver of isotropic mesh adaptation; in particular when the number of element increases.

To increase the slope of convergence, we use *anisotropic* mesh adaptation. In fact, it has been shown in a number of works (for example in [35]) that the Hessian based anisotropic mesh adaptation leads to a convergence of order 2. This fact is highlighted in our results as we observe the increase of the slope going from isotropic to anisotropic mesh adaptation. In Figure 4.5 and 4.6, we observe that our new anisotropic local error indicator $\eta_{\Omega_e,new}$ behaves as expected. This is the first important result of the present work. In fact, we observe that for coarse meshes (i.e. $\mathcal{N} = 2\,000$ elements), the effect of the subscales error estimator is important. This reduction is even slightly better for the case with $a = 10^{-4}$. On the latter figure, we observe that, with only 20 000 elements, we are below the error of an interpolation based anisotropic mesh adaptation of 100 000 elements.

As the number of elements increases, we notice that the effect of the subscales error estimator becomes less important. Indeed, the slopes of the *Anisotropic mesh 2* and *Anisotropic mesh 3* cases decreases. This behavior is expected when we acknowledge that the subscales error estimator can also be considered as an estimation of the subscales error. Thus, when the number of element increases, the local mesh size decreases and, for this reason, the modeled part of the solution becomes smaller.

The second important result of the present work concerns the last (*Anisotropic mesh 3*) case corresponding to anisotropic mesh adaptation taking into account the error estimator computed with bubble functions. Again, we can state that this error estimator is the best choice to drive anisotropic mesh adaptation in comparison with the one computed with stabilizing parameters.



(a) $N = 21\,000$ (b) $N = 21\,927$ (c) $N = 20\,798$

(d) $N = 20\,932$ (e) $N = 23\,834$ (f) $N = 23\,182$

Figure 4.4 – Error distributions and meshes of 2D regular boundary layers for different mesh adaptation techniques

In addition, we notice that the subscales error estimator computed with stabilizing terms has no effect when we reach an important number of elements (for example: $\mathscr{N} = 100\,000$).

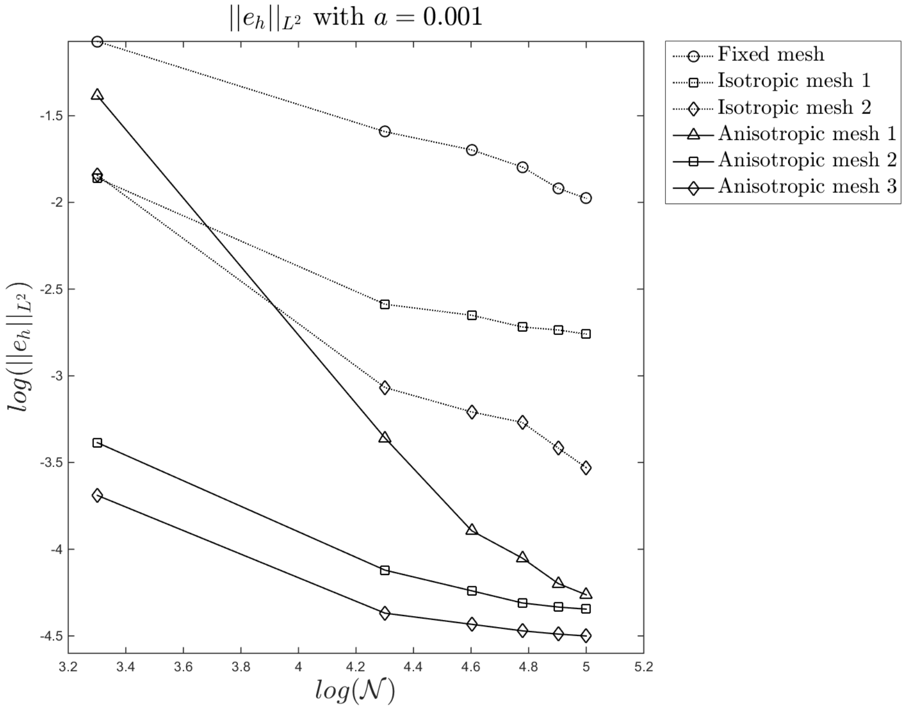


Figure 4.5 – Error in L_2 norm with $a = 10^{-3}$ for regular boundary layers in 2D

4.4.2 Case 2: parabolic boundary layers with recirculating convective field

For this benchmark, we consider a domain $\Omega = (-1,1)^2$ with a non-constant flow. The diffusion coefficient is $a = 0.001$. We apply a zero source term inside the domain. The right side wall has a Dirichlet boundary condition equals to 1. The other walls have a homogeneous Dirichlet boundary conditions. We apply a recirculating convective field defined by $\mathbf{v} = (2y(1-x^2), -2x(1-y^2))$ that will form boundary layers in the domain. The SUPG solution is given in Figure 4.7.

This example represents the propagation of the right wall heat inside the domain through the convective field. For this complex problem, it is difficult to capture the structure of the boundary layer. Our goal here is to show the efficiency of our new anisotropic error indicator in the capture of the parabolic layer. To illustrate the latter, we plot the two different error estimators (one computed with stabilizing parameter and the other with bubble functions) in Figure 4.8. The aim of this figure is to compare the two isotropic components of the two different error estimators. Therefore, it is sufficient to plot only the subscales part of the error estimation. As a result, we observe that the localization of the error is improved by the bubble functions method.

The resulting adapted meshes are presented in Figure 4.9. The cases (a), (b), (c), (d), (e) and (f) correspond to the same mesh adaptation techniques than in the previous example (Case 1). In Figure 4.9 (b) and (e) we observe that the isotropic and anisotropic mesh adaptation techniques follow the localization prescribed by the error estimator based on the stabilizing parameter (see Figure 4.8). Consequently, the mesh is adapted only on a restraint region of the parabolic boundary layer. In Figure 4.9(d), we observe that the conventional Hessian based anisotropic mesh adaptation already capture the 4 boundary layers. However, it does not take into account the sub-grid information that can improve the anisotropic mesh adaptation. In fact, the mesh adaptation tech-

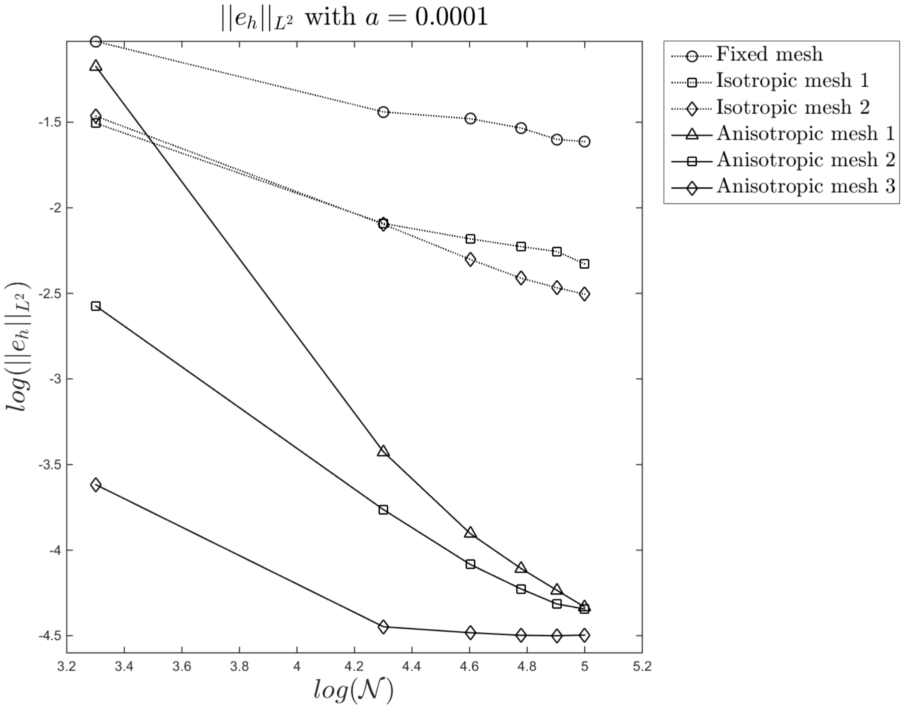


Figure 4.6 – Error in L_2 norm with $a = 10^{-4}$ for regular boundary layers in 2D

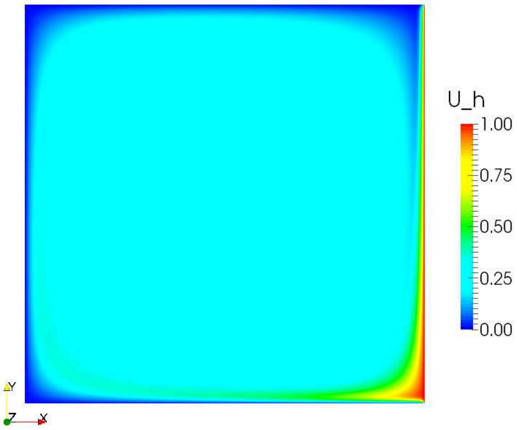


Figure 4.7 – Numerical SUPG solution for parabolic layers with $a = 10^{-3}$

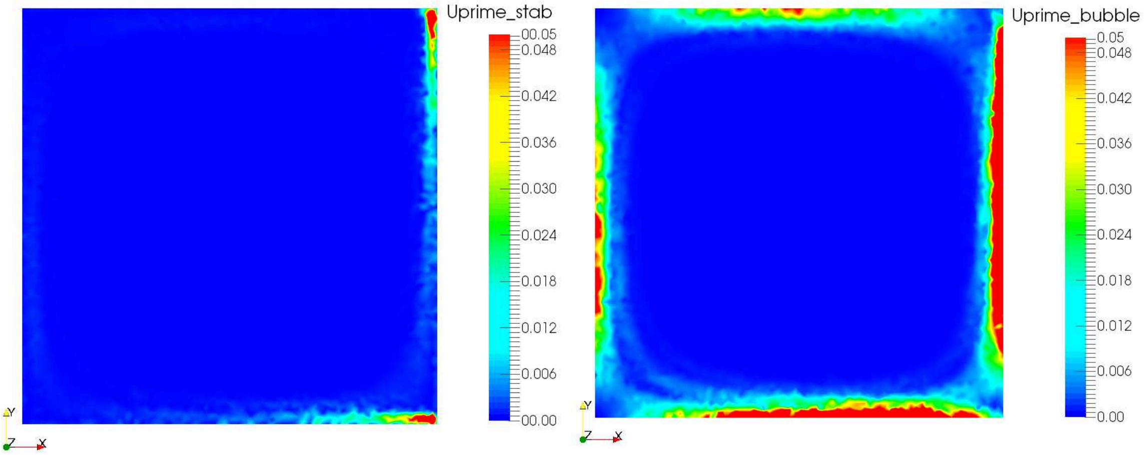


Figure 4.8 – Subscales error estimator computed with stabilizing parameters (left) and bubble functions (right)

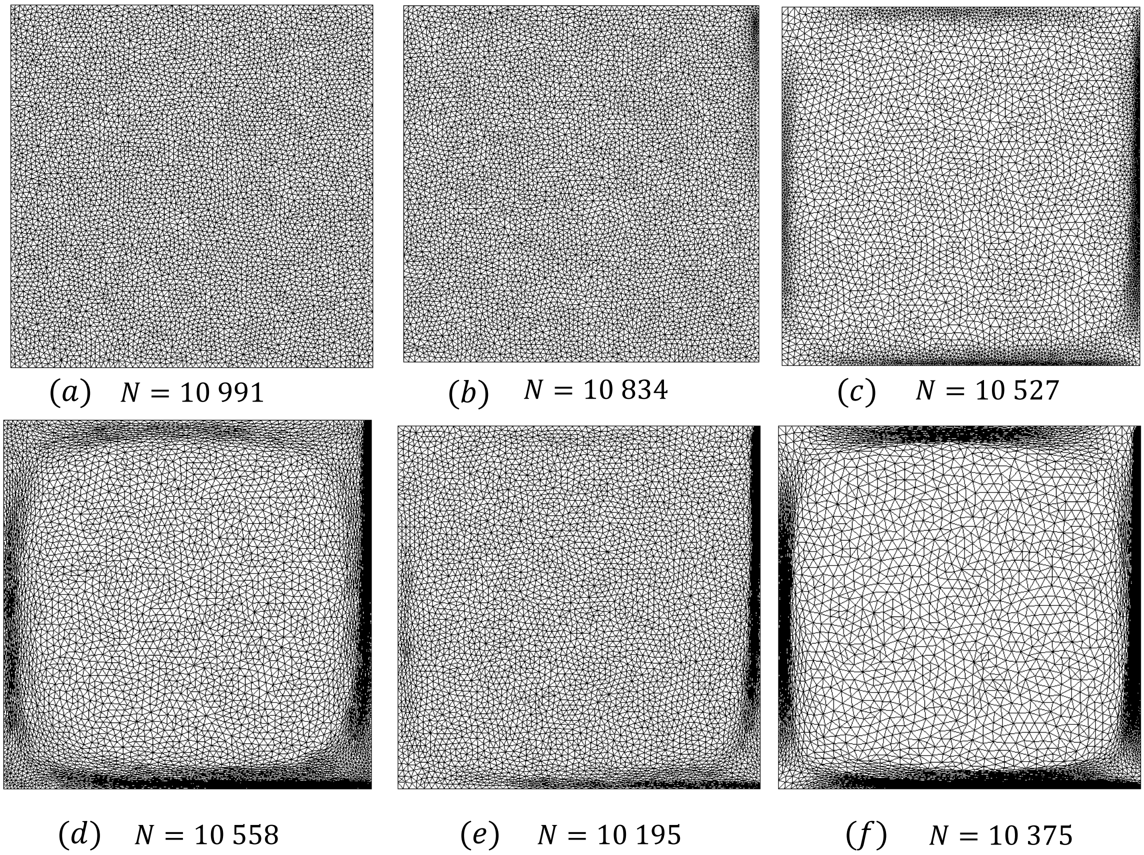
nique using the *new* anisotropic local error indicator taking into account the subscales error indicator based on bubble functions (see Figure 4.9(f)) shows to be the best choice to accurately capture the 4 parabolic boundary layers.

4.4.3 Case 3: regular boundary layers in 3D

This 3D benchmark case is inspired by the one in Section 4.4.1. It takes the same configuration concerning velocity field, source term, boundary conditions and analytic solution with an extrusion in the z direction. However, the convection-diffusion equation is now solved on a cubic domain $\Omega = (0,1) \times (0,1) \times (0,1)$. The analytic solution is given in Figure 4.10. We compute the FE SUPG solution u_h with the 6 mesh adaptation techniques previously describe for 9 mesh sizes: $\mathcal{N} = 40\,000,\ 80\,000,\ 120\,000,\ 160\,000,\ 200\,000,\ 300\,000,\ 400\,000,\ 500\,000$ and $600\,000$ elements.

As for the 2D benchmark, the error distribution corresponding of each mesh is given in Figure 4.11. In Figure 4.11.(a), (b) and (c), we observe the instabilities due to the highly convective regime. Having isotropic elements, the local mesh size in the direction of the velocity field is too important. As said in the introduction of this chapter, we observe that anisotropic mesh adaptation is a really efficient technique to eliminate the spurious oscillations in the boundary layer. This is highlighted in Figure 4.11.(d). However, there is still a part of the error that is not reduced on this figure. To counter this effect, we see that our new anisotropic local error indicator $\eta_{\Omega_e,new}$ is, again, efficient. Also, we observe that the constraint of a fixed number of elements is again well respected with a deviation of less than 6% with the targeted $\mathcal{N}$. Finally, the analysis on error distribution in 2D still holds in 3D and moreover, it seems to be enforce by adding a new dimension to the problem.

The above statements on error distribution are also to be found in the global L^2 norm of the error $||e_h||_{\mathrm{L}^2}$ in Figure 4.12. Moreover, the effects of the subscales error estimator on anisotropic mesh adaptation is enforced in 3D. The important result here is that, with only 40 000 elements in 3D, the error is below an interpolation based anisotropic mesh adaptation on 600 000 elements with our new anisotropic local error estimator $\eta_{\Omega_e,new}$. Again, taking into account the sub-scale information in mesh adaptation makes it possible to reduce drastically the error on the 3D coarse meshes. With about 20 times less elements, we obtain the same global error $||e_h||_{\mathrm{L}^2}$.



(a) $N = 10\,991$ (b) $N = 10\,834$ (c) $N = 10\,527$

(d) $N = 10\,558$ (e) $N = 10\,195$ (f) $N = 10\,375$

Figure 4.9 – Resulting meshes for different types of mesh adaptation for the parabolic boundary layers.

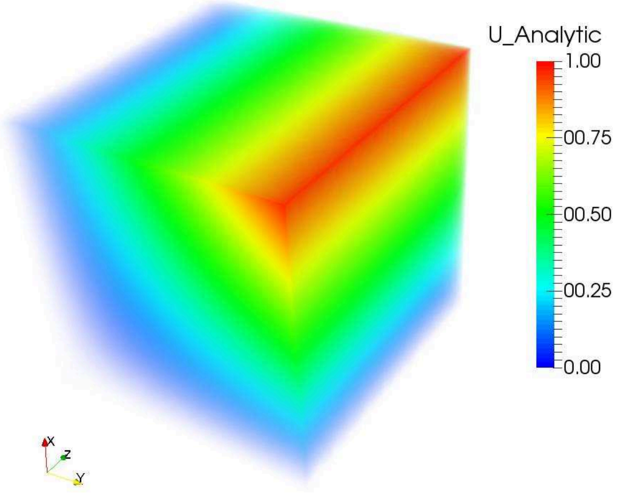
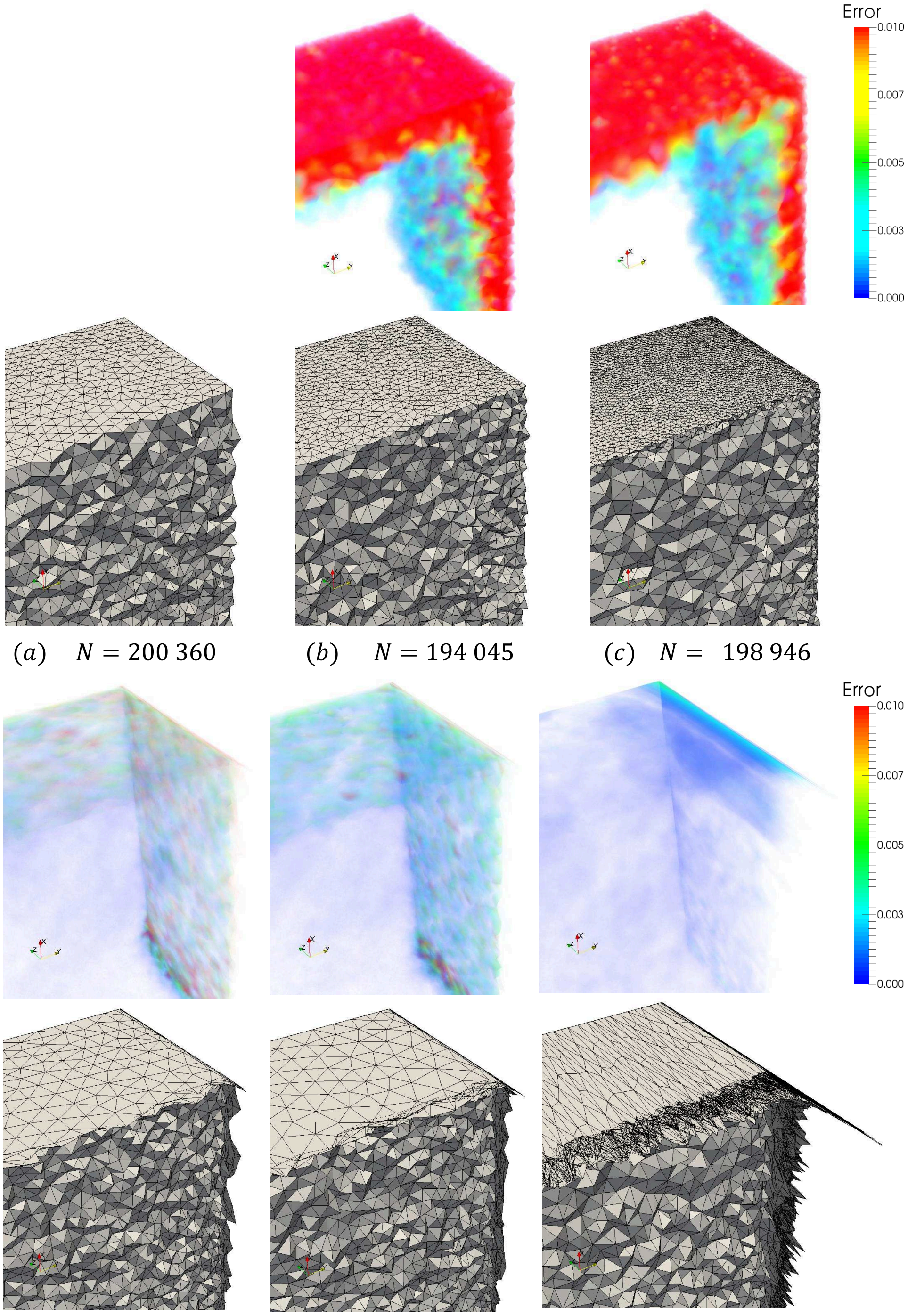


Figure 4.10 – Analytic solution for 3D convection-diffusion equation with $a = 10^{-3}$



(a) $N = 200\ 360$ (b) $N = 194\ 045$ (c) $N = 198\ 946$

(d) $N = 212\ 242$ (e) $N = 200\ 933$ (f) $N = 199\ 659$

Figure 4.11 – Error distributions and meshes of 3D regular boundary layers for different mesh adaptation techniques

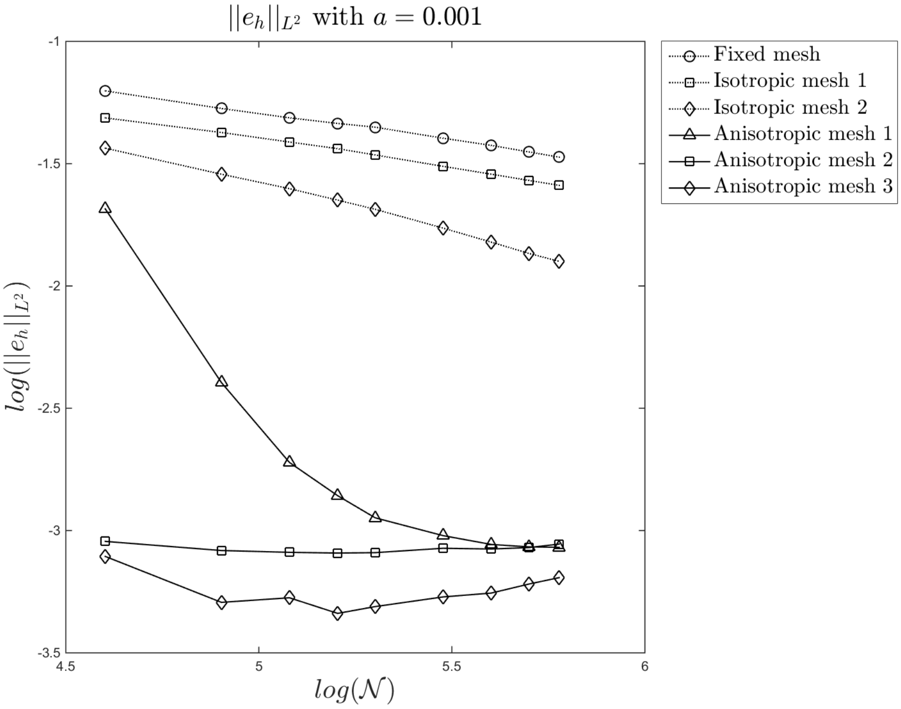


Figure 4.12 – Error in L_2 norm with $a = 10^{-3}$ for regular boundary layers in 3D

4.5 Application to the single impingement jet cooling

On one hand, in the previous section, the new multiscale mesh adaptation techniques were validated on numerical benchmarks with analytic solutions; where an exact error analysis was feasible. On the other hand, in this last section, we propose to apply these new techniques to the industrial case of the single impingement jet cooling. In this case, we cannot compute the exact error because we cannot know a priori the exact solution of the temperature u in the flow. However, the idea here is still to test these mesh adaptation techniques on a realistic aerothermal problem and to present some observations on the resulting meshes. These observations are based on the flow analysis given in Chapter 2 Section 2.3.5. As we will see later, some conclusions about the qualitative efficiency of these new multiscale mesh adaptation techniques can be made.

To realize this test, we propose to use the three types of mesh adaptation techniques corresponding to the previous section and recalled here:

- ***Anisotropic mesh 1:*** Using the anisotropic metric tensor $\mathcal{H}_{aniso}$ computed *without* the subscales error estimator,
- ***Isotropic mesh 2:*** Using the isotropic metric tensor $\mathcal{H}_{iso}$ computed with the subscales error estimator using the bubble functions,
- ***Anisotropic mesh 3:*** Using the new anisotropic metric tensor $\mathcal{H}_{aniso}^{new}$ computed with the subscales error estimator using the bubble functions.

In this realistic aerothermal problem, the unsteadiness of the flow has to be taken into account. Therefore, the mesh adaptation techniques need to be dynamic methods. It means that the mesh is dynamically adapted over time during the simulation. We choose a mesh adaptation step of 10 increments. Referring to Chapter 2, we have 5000 simulation increments. Therefore, we realize 500 mesh adaptations steps for the total simulation time. Doing so, we allow the mesh to follow the flow dynamics over time. The mesh adaptation techniques are tested on a 2D and a 3D configuration of the problem.

The results of the 2D configuration are given in Figure 4.13. In this figure, we observe that the proposed subscales error estimator $\frac{||u'||_{L^\infty(\Omega_e)}}{u'_{\mathrm{TOL}}}$ allow us to locate with good accuracy the unresolved part of the thermal solution. Then, thanks to our new anisotropic mesh adaptation technique, the mesh is adapted anisotropically according to this sub-scale information. Doing so, it allows capturing accurately the turbulent vortex structures in the flow. Indeed, as we can see on the right part of Figure 4.13, this dynamic multiscale mesh adaptation technique allows us to follow the secondary vortices over time during the unsteady simulation.

In Figure 4.14, we compare the different mesh adaptation techniques for a fixed number of elements $\mathcal{N}$ equals to 400 000. In this figure, we observe that the two new multiscale techniques (Isotropic 2 and Anisotropic 3) show important differences with the standard one (Anisotropic 1). In fact, we observe that the mesh follows more closely the thermal solution and in particular, it follows the convective heat exchanges due to the appearance of the secondary vortices. Furthermore, we observe on the zoom view of Figure 4.14 that the new multiscale anisotropic mesh adaptation (using the metric tensor $\mathcal{H}^{new}_{aniso}$) is, again, the choice that gives us the most suitable mesh regarding the complexity of the flow behavior.

We can also observe in this figure that the mesh sizes are slightly different between the three meshes. In fact, the re-meshing tool is based on an optimization procedure on unstructured meshes. The solution is thus not unique and therefore, the targeted number of elements can be different from an iteration to the other. However, the difference between the meshes represents less than 15% of the total mesh sized. Therefore, we consider that the comparison between meshes still holds.

The results for the 3D configuration are given on Figure 4.15. We start the mesh adaptation procedure from an initial isotropic mesh of 800 000 P1 elements. A constrain of 4 000 000 P1 elements is imposed to the optimization problem. The resulting adapted meshes are presented in Figure 4.15. They corresponds to the meshes at increment I = 900 i.e. when the secondary vortices just start forming. They counts 4 263 472 P1 elements for the Anisotropic mesh 1 and 4 340 428 P1 elements for Anisotropic mesh 3. It gives us a gap of less than 10% of the targeted value and we consider that the two meshes are comparable in terms of numerical results.

In Figure 4.15 (top left), we observe that the anisotropic hessian based adapted mesh (Anisotropic mesh 1) is indeed refined according to the anisotropic variations of temperature. We also observe that the temperature is well convected by the velocity field as we recognize the specific turbulent vortex structures, highlighted in Chapter 2. However, in Figure 4.15 (top right), we observe that the multiscale isotropic mesh adaptation (Isotropic mesh 2) captures the flow structures but require an important amount of elements on the plate to do so. This is due to the fact that we have a highly directional flow and that isotropic mesh adaptation is, in any case, not adapted in this situation.

Finally, it is clearly visible that the hessian based mesh adaptation (Anisotropic mesh 1) does not fully capture the complex vortex structures near the plate. It is apparent when we compare the two adapted meshes on the plate plane (see Figure 4.15 (top left and bottom left)). In fact, we observe that the new multiscale mesh adaptation (Anisotropic mesh 3) allows capturing more in details the complex flow behavior on the plate. We recognize the kind of structure that we obtained before for the velocity field in Figure 2.14.

Furthermore, we add in Figure 4.16, a slice view of the meshes on the plate supporting the velocity magnitude field. In this figure, we observe that the new multiscale anisotropic adapted mesh (Anisotropic mesh 3) is the one that is best fitted to the velocity field. Therefore, from these

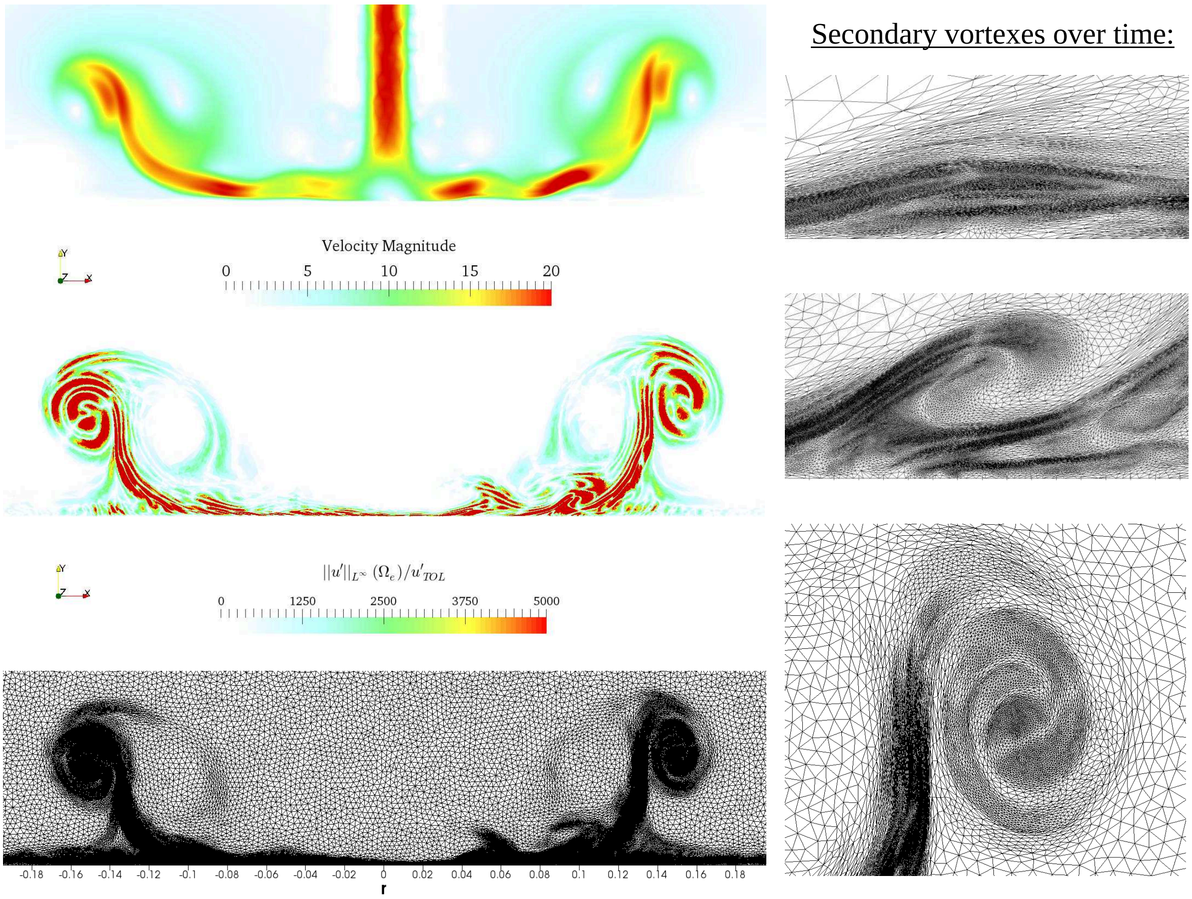
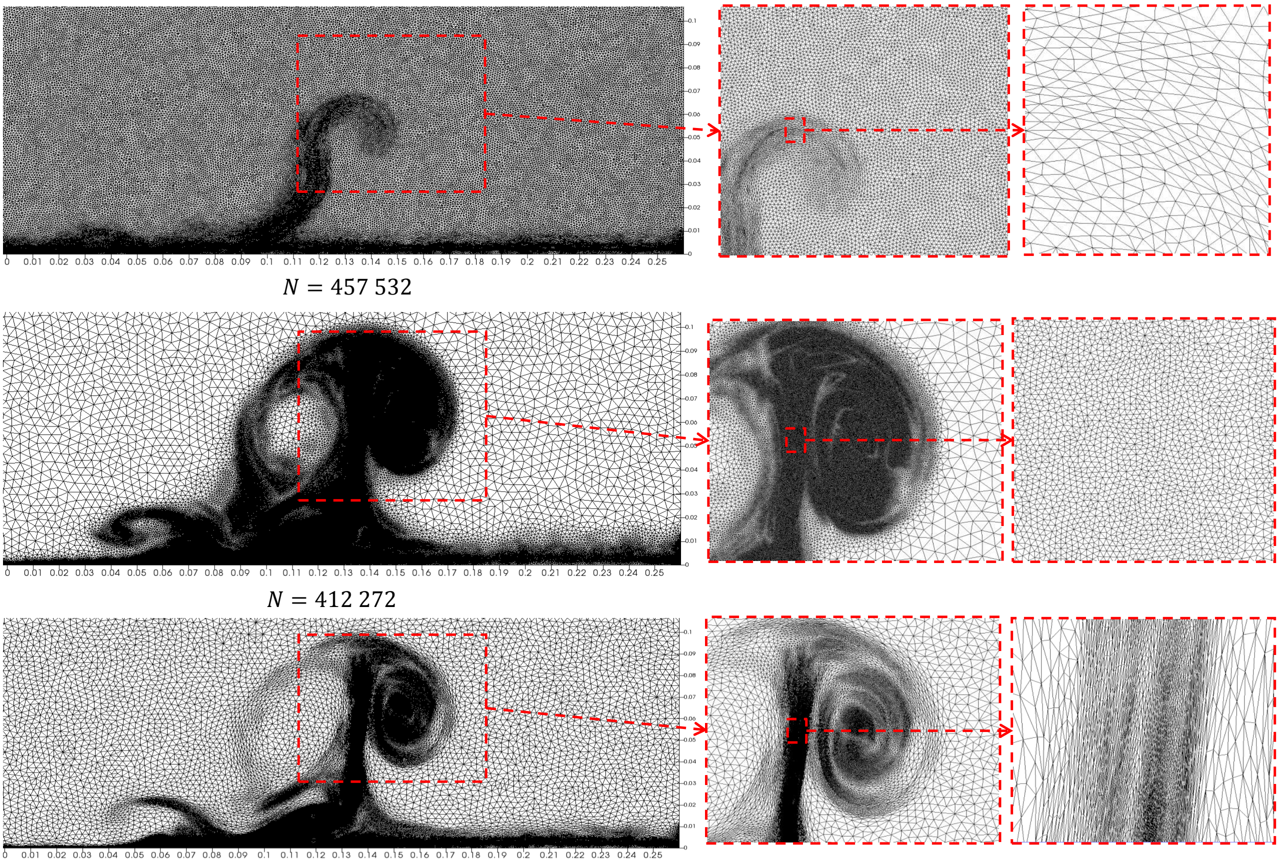


Figure 4.13 – Localization of the subscales error and new multiscale anisotropic mesh adaptation



$N = 457\ 532$

$N = 412\ 272$

$N = 443\ 493$

Figure 4.14 – Resulting meshes for different mesh adaptation techniques in 2D. (top) Anisotropic mesh 1: anisotropic Hessian based mesh adaptation; (middle) Isotropic mesh 2: new isotropic mesh adaptation based on the subscales error estimator; (bottom) Anisotropic mesh 3: new anisotropic mesh adaptation based on the subscales error estimator.

observations, we can expect the new multiscale anisotropic mesh adaptation to be better suited to capture the complex convective heat transfer on the plate. Furthermore, we observe on this view that the mesh is well adapted on the plate. Therefore, the thermal gradient and thus the heat transfer should be well captured by our new anisotropic multiscale mesh adaptation technique.

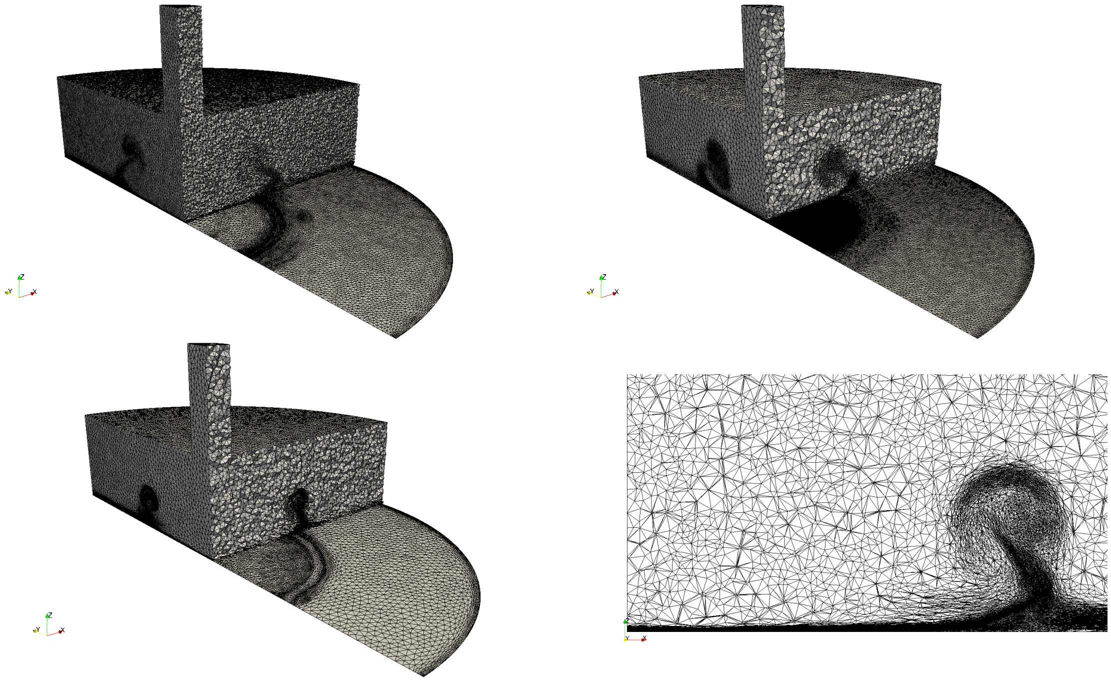


Figure 4.15 – Resulting meshes for different mesh adaptation techniques in 3D. (top left) Anisotropic mesh 1: anisotropic Hessian based mesh adaptation; (top right) Isotropic mesh 2: new isotropic mesh adaptation based on the subscales error estimator; (bottom left) Anisotropic mesh 3: new anisotropic mesh adaptation based on the subscales error estimator; (bottom right) Zoom view of Anisotropic mesh 3.

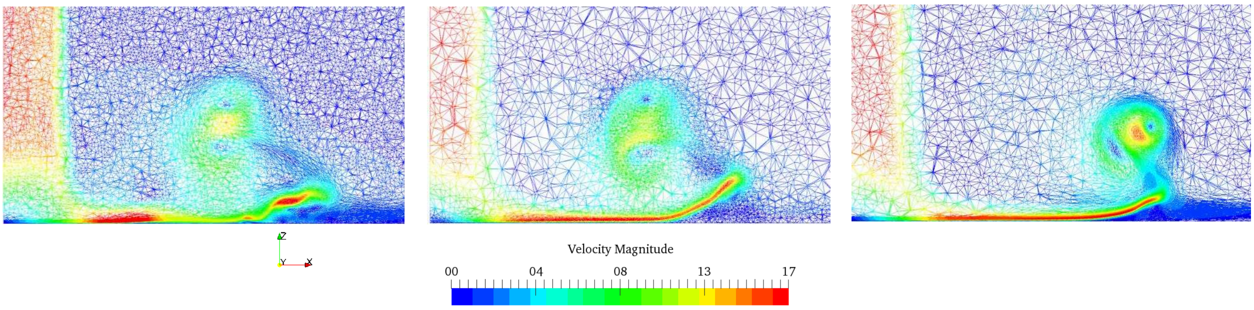


Figure 4.16 – Slice view of resulting meshes for different mesh adaptation techniques in 3D: (left) Anisotropic mesh 1; (middle) Isotropic mesh 2; (right) Anisotropic mesh 3.

4.6 Conclusion

In this chapter, we proposed a new a posteriori error estimator based on the VMS method for anisotropic adaptive fluid mechanics problems. This new error estimator was defined by the combination of both (i) an interpolation based anisotropic error indicator and (ii) a subscales error estimator. The subscales error estimator was computed using two different methods. The definition of a new re-meshing optimization problem allowed us to include this sub-grid information in mesh adaptation with respect to the constraint of a fixed number of elements.

We first validated these multiscale mesh adaptation techniques on 2D and 3D numerical benchmarks, where an exact error analysis was feasible. The results showed that, for the subscales error estimator, the method using bubble functions was better suited to drive both isotropic and anisotropic mesh adaptation. In fact, the localization of the residual subscales error was better established with this method. Then, the results showed that the proposed multiscale mesh adaptation technique allowed us to compute highly precise solutions with much less elements in comparison with other mesh adaptation techniques. Finally, this work showed that this new multiscale anisotropic mesh adaptation strategy was capable of dealing with boundary layers of convection-diffusion problems associated with high Peclet number flows in 2D and 3D.

We then tested these multiscale mesh adaptation techniques on the realistic industrial case of the single impingement jet cooling. This time an exact error analysis was not possible, but some qualitative conclusions could be made. Indeed, the results showed that the combination of the VMS error estimator with anisotropic mesh adaptation allowed getting relevant anisotropically adapted meshes that captured the complex flow structures, which are the secondary vortices generated by the impinging jet. Indeed, the new multiscale anisotropic adapted mesh was better fitted to the solution. It is therefore a promising idea to simulate complex CFD aerothermal problems.

The next steps of this work is first to demonstrate quantitatively the efficiency of this new multiscale mesh adaptation method on the realistic industrial case of the impinging jet cooling. To do so, we propose as a perspective, a comparison of the obtained heat transfer distribution on a large fixed mesh, a hessian based adapted mesh, and a multiscale adapted mesh. In fact, as soon as the physical modeling issues on the impingement jet cooling are dealt with, it will be interesting to compare the heat transfer distributions obtained on both (i) a multiscale adapted mesh and (ii) a large isotropic reference mesh.

Another possible perspective is also to apply this new multiscale mesh adaptation technique to other equations. Indeed, the subscales error estimator is PDE-dependent and in this work, we applied it to the thermal convection-diffusion equation. However, it also possible to apply this error estimate to the incompressible Navier-Stokes equations. This time, the error estimator is computed on the vector variable $\mathbf{v}$ in order to have an estimation of $||\mathbf{v}'||$. The same was also tested on the variable p in order to have an estimation of $||p'||$. The computation of the error estimate using the residual free bubble matrix have yet to be undertaken for these equations.

In the next chapter, we will present the parallel computational framework in which these numerical mesh adaptation methods were developed. Indeed, the efficiency of these numerical methods is closely linked to the parallel performance of the mesh partitioning and load balancing algorithms used on different parallel computational systems.

Résumé du chapitre en français

Nous proposons dans ce chapitre un nouvel estimateur d'erreur a posteriori basé sur la formulation VMS pour l'adaptation de maillages non-structurés. Ce nouvel estimateur d'erreur est appliqué à l'équation thermique de convection-diffusion associée aux problèmes de Mécanique des Fluides. Dans un premier temps, nous proposons deux méthodes permettant de le calculer. La première méthode se base sur le calcul des termes stabilisants dérivés de l'analyse VMS et la définition de normes locales définies sur les éléments. La seconde méthode en revanche, est basée sur l'utilisation d'une combinaison linéaire de fonctions bulles pour établir un calcul de l'erreur en tous points de l'élément.

La première contribution de ce travail est ensuite de proposer une nouvelle méthode d'adaptation de maillage isotrope basée sur ce nouvel estimateur d'erreur. Pour ce faire, nous définissons un tenseur métrique isotrope appelé $\mathcal{H}_{iso}$ et nous résolvons un problème d'optimisation sur les tailles de mailles permettant de contraindre le maillage à un nombre fixe d'éléments. Jusqu'à maintenant, les estimateurs d'erreur VMS étaient surtout utilisés pour l'adaptation isotrope de maillages. Dans ce travail, nous proposons également d'utiliser cette information de sous-maille dans l'adaptation anisotrope du maillage.

La seconde contribution est donc de combiner à la fois (i) l'indicateur d'erreur d'interpolation et (ii) l'estimateur d'erreur sous-échelle dans le but de réaliser une adaptation anisotrope du maillage. D'après ce que nous avons pu lire dans la littérature, cette stratégie n'a jamais été tentée par le passé. Pour réaliser cela, nous définissons un nouveau tenseur métrique anisotrope appelé $\mathcal{H}_{aniso}^{new}$ qui permet de prendre en compte (i) les variations anisotropes de la solution mais également (ii) l'information de sous maille. En effet, l'approche VMS nous permet de quantifier a posteriori l'erreur de sous maille que nous pouvons ensuite utiliser pour enrichir l'indicateur d'erreur anisotrope des approches hessiennes conventionnelles. Cette nouvelle méthode nous permet donc de tirer partie (i) des méthodes d'adaptation de maillage construites a priori, et (ii) de l'analyse VMS réalisée a posteriori.

Ces deux nouvelles méthodes d'adaptation de maillage multi-échelles sont d'abord testées sur des cas analytiques 2D et 3D où une analyse de l'erreur exacte est faisable. Les résultats montrent que, pour l'estimateur d'erreur sous-échelle, la méthode de calcul utilisant les fonction bulles est la mieux adaptée pour piloter l'adaptation isotrope et anisotrope du maillage. En effet, la localisation de l'erreur de sous-maille est mieux établie avec cette dernière. De plus, les résultats montrent que ces deux nouvelles méthodes d'adaptation de maillage multi-échelles permettent d'obtenir des solutions numériques hautement précises avec bien moins d'éléments en comparaison avec les méthodes conventionnelles. Finalement, ce travail montre que ces méthodes d'adaptation de maillage multi-échelles sont capables de gérer les couches limites associées aux problèmes de convection-diffusion à hauts nombre de Péclet en 2D et en 3D.

Ensuite, ces nouvelles méthodes d'adaptation de maillage sont appliquées à un cas industriel plus réaliste qui est le cas du refroidissement par jet impactant. Cette fois, une analyse exacte de l'erreur n'est bien sûr pas possible mais certaines conclusions qualitatives peuvent quand même être énoncées. En effet, les résultats montrent que la combinaison d'un estimateur d'erreur VMS avec l'adaptation anisotrope de maillage nous permet de capturer les échanges thermiques complexes générés par les structures turbulentes de l'écoulement. En effet, nous montrons que ces nouveaux maillages permettent notamment de suivre les tourbillons secondaires générés par le jet impactant et que le maillage semble être mieux aligné pour capturer les échanges thermiques dans la couche limite. C'est donc une idée prometteuse pour améliorer la simulation de ce cas aérothermique complexe.

Les perspectives de ce travail sont d'abord de démontrer quantitativement les effets de ces nouvelles méthodes d'adaptation de maillage multi-échelles sur des cas industriels plus réalistes tel que celui du refroidissement par impact de jet. Cela sera rendu possible par une comparaison avec des références numériques et expérimentales des résultats de transferts thermiques calculés en utilisant ce type de méthode. Une autre perspective possible de ce travail est également d'appliquer ces nouvelles méthodes multi-échelles à d'autres équations. En effet, l'estimateur d'erreur sous-échelle que nous développons est dépendant de la PDE. Dans ce travail, nous proposons de l'appliquer à l'équation thermique de convection-diffusion mais d'autres équations peuvent être envisagées. Pendant cette thèse, nous avons commencé à nous intéresser à l'application de cette méthode aux équation de Navier-Stokes incompressibles.

Dans le chapitre suivant, nous présentons le contexte numérique parallèle dans lequel ces méthodes numériques d'adaptation de maillage sont développées. En effet, l'efficacité de ces méthodes numériques est intimement liée à la performance en parallèle des algorithmes de repartitionnement de maillage et de répartition de la charge de calcul utilisés sur des systèmes de calcul parallèle.

Chapter 5

Massively Parallel Computational Framework

Contents

5.1 Introduction

In order to produce reliable aerothermal simulations in modern aircraft engines, the future computational systems will have to break out from the existing ones. Indeed, referring to NASA report by Slotnick et al. in [10], considering the actual evolution of technologies, scientists predict that the future systems will reach a computational peak capacity of 30 ExaFLOPS in 2030.

However, several factors show that these exascale systems won't be a simple evolution of the existing petascale systems. In fact, from an energetic stand-point, the existing petascale systems require between 10 to 20 MW, corresponding approximately to 1MW per PetaFLOPS. Following this trend, the theoretical energy consumption of an exascale system will be around 1 GW. For environmental reasons, it is of course not feasible to build such systems. Indeed, the targeted energy consumption for these systems is between 20 to 60 MW. Consequently, it is primordial for computer scientists to find new hardware systems capable to sustain this computational power without prohibitive energy consumption.

Furthermore, to improve the performances of the existing systems, computer scientists cannot rely on increasing the performance of a single core as they used to. For this reason, as mentioned in [91], in the last years, the number of cores per chip and per node have increased quite fast but this aggregation necessitates research work on more complex memory hierarchies, networks topologies and partitioning algorithms.

Finally, these new hardware systems will have to be robust and in particular, more fault tolerant than the existing ones in order to anticipate any kind of failure from a system's component. In fact, by increasing the number of components like the Central Processing Unit (CPU)s for example, the risk to get a failure from one of them is increased. In this case, it is necessary for the computation to continue and for the computational loads to be redistributed efficiently on the other working components.

In this chapter, we present the parallel computational framework in which this PhD work was realized. To do so, we first present the parallel software capabilities of the finite element code Cimlib-CFD and then, the two different hardware systems on which our developments were tested. In Section 5.2, we present the FE framework and the mesh partitioning principles. Then, the algorithm behind parallel mesh adaptation is introduced as well as important notions on the particular treatment of the partitions interfaces. Additionally, we present the method used in Cimlib-CFD to realize load balancing during the mesh adaptation process. To finalize the definition of our parallel framework, a mention of the parallel visualization tools used during this thesis is proposed.

In Section 5.3, we present the two different hardware systems tested during this PhD thesis. In particular, we introduce the first trials on running Cimlib-CFD on a massively parallel supercomputer allowing simulations over a thousand cores. Taking into account the characteristics of the supercomputer, a fault tolerant checkpoint-restart procedure, implemented during this work, is presented. In the last section, applications of the parallel components are proposed. In particular, parallel mesh adaptation is applied to the single impingement jet cooling. Finally, we present the massively parallel computation of a turbine vane composed by 39 holes, launched on the supercomputer along with preliminary results on this geometry.

5.2 Parallel software components in Cimlib-CFD

5.2.1 Finite elements computations

The FE method is used to solve numerically a PDE defined on a domain for a set of boundary and initial conditions. For time-dependent equations, the different techniques used to solve the problem are classified in three categories: (i) explicit, (ii) semi-implicit and (iii) implicit. For incompressible fluid mechanic problems, we usually prefer to use an implicit time discretization for the Navier-Stokes equations due to the following reasons:

- for explicit methods, when the diffusive term is important, the stability criteria on the time step can be very restrictive,
- by using the FE method, even if an explicit time discretization is used, there is still a mass matrix to inverse in order to solve the problem.

The implicit resolution of a PDE using the FE method leads to the solving of large linear or non-linear systems. For a non-linear system, the application of the Newton-Raphson method leads itself to the resolution of several large linear systems. A linear system can be written in matrix form as follows:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{5.1}$$

where **A** is a matrix supposedly invertible, **x** is a vector of unknowns and **b** the second term.

In most involving finite element codes, solving the linear system is done by using iterative procedures. For elliptic problems, the preconditioned Conjugate Gradient (CG) algorithm can be used for example. For non-symmetric problems, the Generalized Minimal RESidual (GMRES) method introduced by Saad et al. in [92] is more suitable. The GMRES method gives an approximation $\mathbf{x}_n$ of the exact solution **x** which minimizes the norm of the residual $||\mathbf{A}\mathbf{x}_n - b||$. In our finite element code, we use the Portable Extensible Toolkit for Scientific computation (PETSc) library to store and solve large systems in parallel. PETSc is an open-source suite of data structures and routines to solve scientific applications modeled by PDEs.

It has been noted in the past like in [10, 11, 93] that, for most implicit simulations, 80% of the computational time is dedicated to the solving linear systems. Furthermore, the size of a linear system grows linearly with the size of the spatial and time dependent domain. This is therefore a critical problematic in FEM and foundational mathematical research in highly scalable linear and nonlinear solvers is still required. In the past, important research work aiming at improving the convergence of iterative solvers using a posteriori error estimates was undertaken, starting with Babuska and Rheinboldt in [94, 95], followed by other authors like in [96–99] for example.

According to Houzeaux et al. in [91], the algebraic solvers are mainly responsible for the limitation of the parallel scalability of finite element codes. The solving of the linear systems is done in parallel using mesh partitioning techniques.

5.2.2 Mesh partitioning

Mesh partitioning is a commonly used technique for parallelization of numerical methods. For several years now, parallelisms by domain decomposition have been well developed for FE methods like in [100–105]. For distributed memory machines, which are currently the most used architecture for large parallel machines, mesh partitioning is used to distribute the workload over the computational domain. From the FE point of view, the problem to solve is subdivided into subproblems and the computational domain into sub-domains. Each sub-domain is thus assigned to a parallel process which carried out the operations on the different fields corresponding to this part of the mesh. An example is given in Figure 5.1. In this figure, each color corresponds to a particular parallel process.

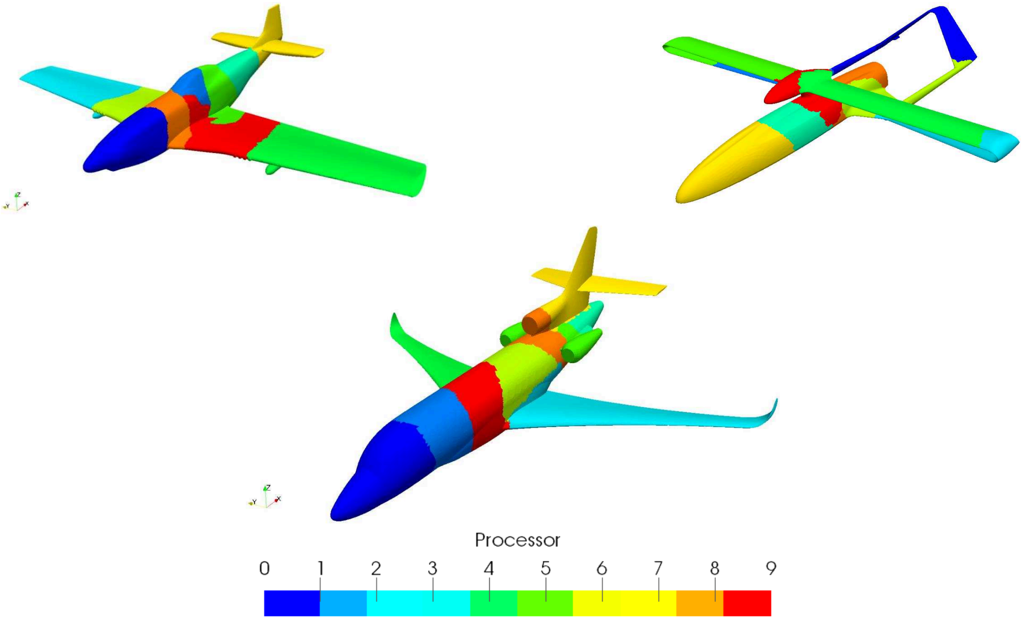


Figure 5.1 – Mesh partitioning on complex 3D geometries: a North American P-51, a Dassault Falcon 7x and a Tyrix Aerospace Albadrone

Different types of operations are then possible on a partitioned mesh. On one hand, some of them require no communications between the processes. For example, the unique value of a norm can be calculated by adding the contributions from all the parallel processes via data recollection. On the other hand, operations like the Sparse Matrix Vector product (SpMV), occupying a central place in iterative solvers, require communications with the neighboring sub-domains. Indeed, they are related to the coupling at the partitions borders and the data need to be exchanged in order to include the interfaces in the computation of the new vector.

Thus, in CFD codes, different types of operations coexist and a unique partition suitable for all of them needs to be generated. This partition have to provide a good balance of the resulting workload, while minimizing the communication requirements. In general, the communication costs are proportional to the size of the subdomains interfaces and therefore, it is necessary to generate partitions that minimize them.

To evaluate the parallel performances of CFD codes, especially for large computations, the notion of scalability is primordial. The scalability of a system is its capacity to handle a growing amount of work, when allocated resources are added. For example, in CFD, a code will be considered scalable if, by adding an increasingly large number of computational resources, like CPUs, the code is capable of dealing with increasingly large problems in terms Degrees Of Freedom (DOF) or number of elements. In the context of HPC there are two common ways to define the scalability of a code:

- the strong scalability: it is defined by how the computational time vary with the number of processors for a fixed total problem size,
- the weak scalability: it is defined by how the computational time vary with the number of processors for a fixed problem size per processor.

The weak scalability is most interesting for O($\mathcal{N}$) algorithms. In this case, perfect weak scaling shows a constant time, independent of processor count, to compute the solution. Deviations from this indicate that the overhead due to parallelism is increasing. This overhead is mainly due to the communication time between the processors and tends to be reduced when we minimize the size of the interfaces between the partitions.

5.2.3 Parallel mesh adaptation

The parallelization of mesh adaptation methods goes back to the end of the 1990s. The Single Program Multiple Data (SPMD) Message Passing Interface (MPI)-based paradigm has been adopted by the pioneering works like in [101, 102, 106] for example. A schematic representation of the algorithm behind parallel mesh adaptation is presented in Figure 5.2.

Initial mesh partitioning and load balancing: At the beginning of the simulation, the initial mesh is partitioned over the allocated resources and an initial load balancing is done based on the mesh topology and on the geometry of the domain.

Read and Solve: Then, the subdomains are read by the program and operations are done locally, i.e. within each partition. The solving step contains the part of the finite element code, which corresponds to the building and the solving of linear systems. Indeed, operations like the solving of the Navier-Stokes VMS system or the thermal convection-diffusion system are realized along with other computations required for the simulation like the computation of the local error estimates for instance.

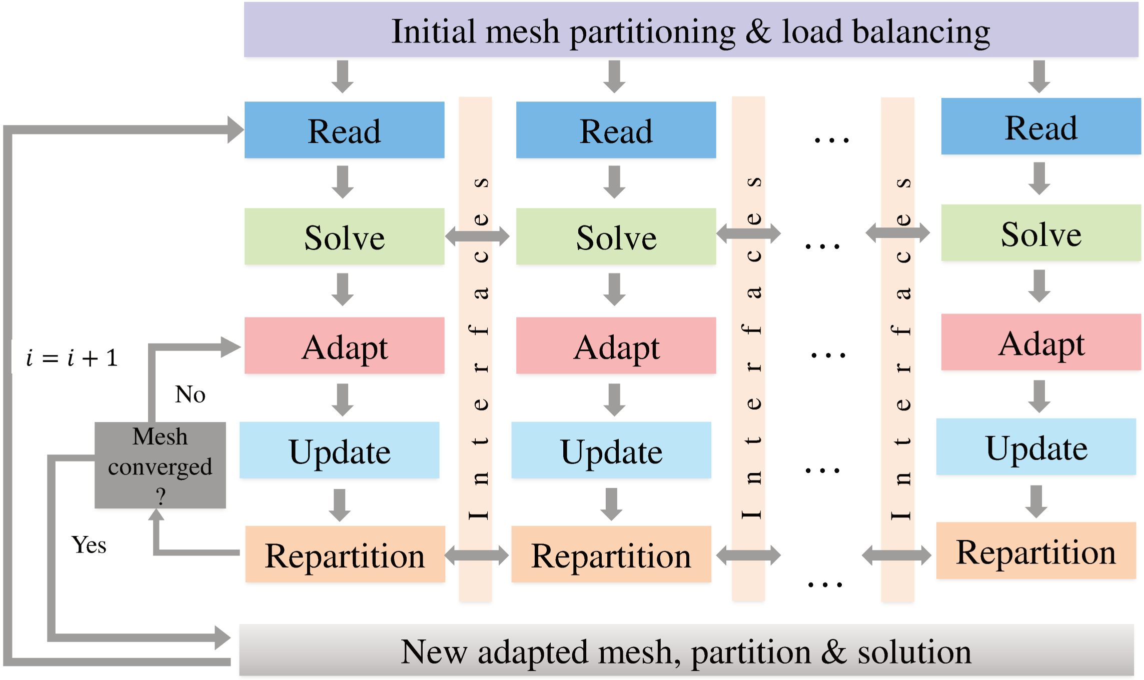

Initial mesh partitioning & load balancing
Read
Solve
Adapt
Update
Repartition
Interfaces
...
$i = i + 1$
No
Mesh converged ?
Yes
New adapted mesh, partition & solution

Figure 5.2 – Illustration of the parallel mesh adaptation algorithm

Adapt: The mesh is then adapted using the mesh adaptation technique presented in Chapter 4. From the local error estimator, we correlate the error to the mesh geometry using a metric space in order to generate a new adapted mesh. Referring to Chapter 4, to adapt the mesh, an error estimate is computed for each subdomain. According to the derived error estimator, and under the constraint of a given number of elements in the new adapted mesh, an optimal mesh is generated. The constraint could be considered as local to each subdomain. In this case, solving the error estimate problem is straightforward. Indeed, all computations are local and there is no need to exchange data between the processors. The local constraint on the number of elements implies the generation of a new mesh with the same number of elements per processor. This allows avoiding heavy load balancing cost after each mesh adaptation.

However, this approach tends towards an overestimate of the mesh density on subdomains where flow activity is almost neglected. From a scaling point of view, this approach leads to a weak scalability model for which the problem size grows linearly with respect to the number of processors. To derive a strong scalability model, which refers in general to parallel performance for a fixed problem size, the constraint on the number of elements for the new generated mesh should be global. The global number of elements over the entire domain is distributed with respect to the mesh density prescribed by the error estimator. This is a hard scalability model that leads to good parallel performances. However, reload balancing is needed after each mesh adaptation stage. The parallel behavior of the mesh adaptation is very close to the serial one and the error analysis is still the same. For these reasons, this model is more relevant than the former one.

Update: The solution computed on the previous mesh then needs to be updated on the new mesh. At this step, we use interpolation methods to transport the variables from the previous mesh to the new one. In the context of anisotropic mesh adaptation with highly stretched elements, this interpolation step is crucial and often leads to some difficulties. Indeed, interpolation methods can reduce the conservation of important physical quantities and leads to errors that spoil the solution accuracy. In the past years, important research work has been done on this subject like in [107–111] for example. In particular, in [112], Bahbah et al. propose a globally conservative methods suitable for both interpolations on unstructured fixed and adaptive anisotropic meshes. It consists in combining an a posteriori error estimator that minimizes the interpolation error of the finite element solution followed by an interpolation with restrictions method that conserves physical properties of the field being interpolated.

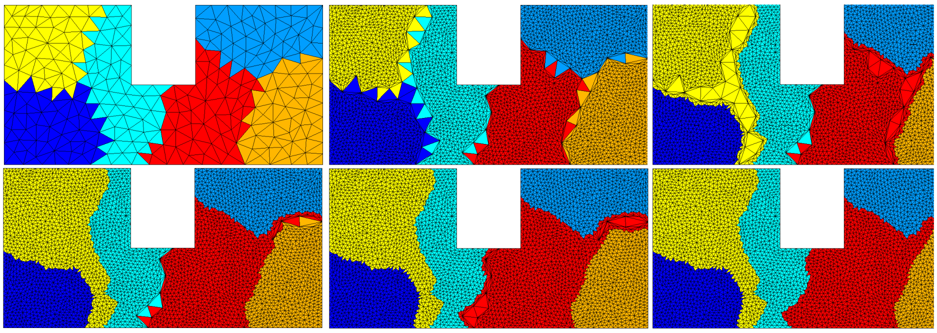


Figure 5.3 – Iterative parallel remeshing steps on a 2D distributed mesh

Repartition: Finally, the new mesh is repartitioned over the allocated CPUs to take into account for the changes of mesh topology in the computational loads distribution. Indeed, mesh repartitioning is the key process of parallel mesh adaptation methods [103, 105]. The effectiveness of the parallel mesh adaptation method depends on the repartitioning algorithms used and on how the interfaces between subdomains are managed. Starting from a partitioned mesh into multiple sub-domains, remeshing operations are performed using a sequential mesh adaptator on each subdomain with an extra treatment of the interfaces. Two main approaches are then considered in the literature:

1. An iterative one: At the first iteration, remeshing is performed concurrently on each processor while the interfaces between sub-domains are locked to avoid any modification in the

sequential remeshing kernel. Then, a new partitioning is calculated to move the interfaces and remesh them at the next iteration. As illustrated in Figure 5.3, the algorithm iterates until all items have been re-meshed.

2. The second approach consists in handling the interfaces by considering a complementary data structure to manage remeshing on remote mesh entities.

The first approach is preferred because of its high efficiency and fully code reusing capability for sequential remeshing kernels. However, the unstructured and dynamic nature of mesh adaptation algorithms leads to imbalance the initial workload. Therefore, an efficient dynamic load balancing procedure is needed after each mesh adaptation step.

5.2.4 Dynamic load balancing

The load balance is one of the principal measurements to quantify the parallel performances of a HPC system. It quantifies the workload distribution between the computational resources involved in a computational task. Referring to [91], if $time_i$ denotes the time spent by process i, out of n_p processes, on the execution of a parallel task, the load balance can be expressed as the averaged time $ave_i(time_i)$, divided by the maximum time, $max_i(time_i)$. This value represents the ratio between the resources effectively used with respect to the resources engaged to carry out the task. It is defined by the following formula:

$$load\ balance := \frac{ave_i(time_i)}{max_i(time_i} \tag{5.2}$$

In this work, we follow the same load balancing strategy than Mesri et al. in [103]. A cost function is defined and takes into account the theoretical computation and communication time of the allocated resources. Then, the load balancing process is realized using two major steps:

- Forming disjoint pairs of processors that are susceptible to minimize the cost function,
- Optimizing the cartography on each pair. This optimization is done by transferring mesh nodes or mesh cells from a processor to the other using the notion of *strip migration*.

These two steps are repeated as long as the global cost of the partition can be optimized. The results from [103] show that the use of this method on various system architectures allows accelerating the mesh partitioning process. In terms of scalability, a linear behavior is observed. Indeed, the mesh partitioning time evolves linearly with the number of processors and mesh nodes which is a proof of good scalability performances.

An example of load balancing is given in Figure 5.4. It is the same example that the one in Figure 5.3 after the parallel remeshing procedure. On the left side of Figure 5.4, it is trivial to observe that the partition is not optimal. In particular, as mention before, the size of the interfaces is too large and this could have a damaging impact on the communication costs. Therefore, the cost function is optimized using the previously described load balancing procedure by transferring nodes from one processor to the other in order to obtain the final optimal partition presented on the right side of Figure 5.4.

5.2.5 Parallel visualization component

In this PhD work, the meshes used can reach several millions of points on which solution data and simulation parameters have to be stored. The file format used to store these results is the Visualization Toolkit Unstructured points data (VTU) format. Because of the large number of data, these files can exceed several gigabytes and they can be challenging to load using a single local machine.

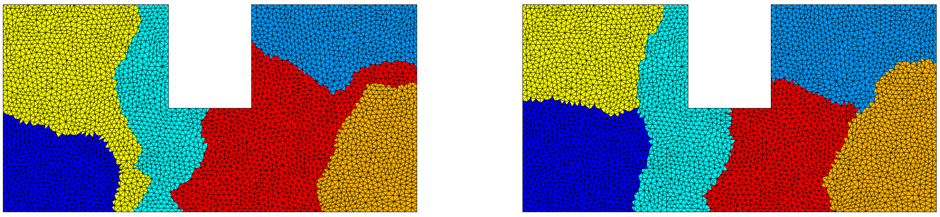


Figure 5.4 – Before (left) and after (right) load balancing

Therefore, parallel visualization tools are necessary in order to reduce the post-treatment computational times required by the visualization program. To realize this, we use the visualization tool ParaView, and we launch it in parallel on the cluster.

ParaView is an open-source, multi-platform data analysis and visualization application. It allows building visualizations to analyze resulting data using qualitative and quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities, implemented using the python language. In particular, ParaView was developed to run on supercomputers in order to analyze extremely large datasets using distributed memory computing resources. To do so, it uses the client-server environment. In this mode, ParaView is launched on a local machine, named client, but all the computations necessary for the visualization are done on the server, i.e. on the cluster.

Doing so, we manage opening the results and realizing the post-treatment analysis in parallel, directly on the Cluster Intel. Furthermore, by visualizing the results in parallel, it allows completing the parallel computational framework of Cimlib-CFD. Indeed, from the mesh partitioning to the post-treatment analysis, we ensure the full parallel capability of our numerical tools.

Note: in this PhD thesis, the client-server strategy have been set up only on the Cluster Intel. As perspective, it would be optimal to set it up on the Occigen II supercomputer. Indeed, it will allow users to use the capabilities of the latter for visualization.

5.3 Hardware systems dedicated to High Performance Computing

Parallelization techniques have been broadly studied on homogeneous hardware architectures where all the CPUs are identical and connected to one another with a fast communication network. However, with the evolution of HPC, hardware architectures are more and more diversified. Indeed, today's supercomputers are usually an agglomeration of clusters, CPUs and communication networks that have their own specific characteristics. Mesri et al. give an example in [103] where the focus is put on Grids. In fact, the Grid concept allows furnishing coordinate computational resources coming from different institutions or organizations for one same simulation. This concept has been studied for several years and have caught the interest of numerical scientists, especially in the domain of CFD. In particular, the authors in [103] show how the architecture characteristics can affect the efficiency of parallel mesh partitioning. For this reason, it is necessary to complete the traditional parallelization schemes to take into account for the diversity of hardware architectures encountered in the numerical scientific world.

In this work, we manage to test our developments and to produce numerical results on two different systems. The first one is an intern system from the CEMEF research center. It is administrated by CEMEF MINES-Paristech and develops a theoretical computational power of 25.8 TeraFLOPS. The second one however, is the GENCI Occigen II supercomputer administrated by the Centre Informatique National de l'Enseignement Supérieur (CINES) in Montpellier. It has a theoretical computational power of 3.5 PetaFLOPS and was ranked 26^{th} on the Top500 list in

November 2014.

The goal of this section is to present the characteristics of these two different systems in terms of hardware environment and scheduler characteristics. Then, we present the work done in the scope of the ARTEMIS project that allowed us to get access to the GENCI Occigen II supercomputer. Finally, we present the implementation of the checkpoint-restart shell scripts used to adapt Cimlib-CFD to the scheduling constrains of the supercomputer.

5.3.1 Parallel computations on the lab's Cluster Intel

In this work, the computations have first been performed on our lab's cluster. It consists of 2000 heterogeneous cores interconnected with infiniband network. The CEMEF's cluster is a linux system with two Intel Xeon generations E5-2670 and E5-2680 chips. The main node characteristics of this platform are described in Table 5.1.

Name	Processor Family	# of cores	Core frequency (Ghz)	Cache Memory (MB)	Memory (GB)
CEMEF 1	Xeon-E5-2670	20	2.6	25	64
CEMEF 2	Xeon-E5-2680	24	2.5	30	64
CEMEF 3	Xeon-E5-2680	28	2.4	30	128

Table 5.1 – Characteristics of the SMP nodes on the Cluster Intel

The jobs are scheduled using the OAR environment presented by Capit et al. in [113]. OAR is a versatile resource and task manager, also called a batch scheduler. It allows managing computing resources on a multi-users and multi-tasks system. Indeed, in this environment, the user can: (i) choose the computational resources needed, (ii) plan his jobs by choosing starting dates and wall-times, (iii) reserve computational resources in order to have exclusivity on them for a certain amount of time. Using this information, the OAR environment schedules the different jobs coming from the different users. The lab's policy allows the user to choose a maximum wall time of 3000 hours (i.e. 125 days, i.e. 4 months) which is largely sufficient for this work's computations. However, the resources requests exceeding 200 cores have to be on nodes reserved in advance to ensure a proper distribution of the computing resources between users.

To choose the computational resources needed for a job, two important things have to be taken into account. First, a trade-off has to be found between using a large number of CPUs and limiting the communications between them. Indeed, for a fixed size problem, a large number of cores may also implies a large number of communication between them. Therefore, an optimum has to be found on a multi-objective cost function taking into account (i) the CPU time and (ii) the communication time. This optimum depends mostly on the operations that are computed during the simulation. As mentioned before, some operations, like the SpMV, require a large amount of communications between the processes. Consequently, if the simulation requires computing a large number of SpMV, it will not be appropriate to allocate a large number of CPUs because the communications costs will be penalizing. On the contrary, if most of the operations are norms, the parallelization will be more efficient and thus, a large number of CPUs can be assigned to the simulation.

The second thing to consider is the memory capacity of the CPUs. In the FE framework, a large number of tables need to be allocated at the beginning of the simulation in order to store the entities on the mesh. Furthermore, in our case, as we solve the Navier-Stokes equations implicitly, we need to store the global sparse matrix of the linear system. To do so, we use advanced numerical storage formats. Indeed, in the case of a sparse matrix, substantial memory requirement reductions can be realized by storing only the non-zero entries. Depending on the number and distribution of the non-zero entries, different data structures can be used but they all yield to huge savings in memory when compared to the basic approach. In our case, the Compress Sparse Row (CSR) format allows obtaining a fast access to the matrix entities and thus facilitates the matrix operations required. This format uses three arrays that respectively contain: (i) the non zero values, (ii) the compressed rows indexes (i.e. only the indexes where non-zero values appears) and (iii) the column indexes.

In our case, as we solve two solvers which are (i) the Navier-Stokes VMS solver and (ii) thermal convection-diffusion solver, operations, like matrix-vector multiplications, requiring a large amount of communication are realized during the simulation. Therefore, after some experiments it was observed that the optimum way to choose the computational resources was to allocate between 50 000 to 100 000 mesh elements per cores.

5.3.2 The GENCI Occigen II supercomputer

To obtain resources on the GENCI Occigen II supercomputer, we had to answer one of the two campaigns launched by CINES in Autumn and Spring following the Demande d'Attribution de Ressources Informatiques (DARI) procedure. This procedure was realized in the scope of the ARTEMIS project carried out by Dr. Y. Mesri. Before presenting the hardware system characteristics of the GENCI Occigen II supercomputer, we propose here to give the main outlines of the ARTEMIS project.

5.3.2.1 The ARTEMIS project

The goal of the ARTEMIS project is twofold. Indeed, it aims at both (i) driving architectural decisions and (ii) adapting algorithms to the next generation of HPC computer architectures. ARTEMIS's particular focus is put on the algorithms that underlie the high-fidelity analysis of aerothermal computations implying heat transfer in turbulent flows. Thus, the ARTEMIS co-design process involves to continually evaluates complex architectural and algorithmic trade-offs aiming ultimately at the design of exascale algorithms that can efficiently leverage them.

Indeed, many important thermal transport questions are accessible only through detailed simulations that span the full range of turbulent scales set by the geometry and the high Reynolds numbers involved. Such simulations require extreme-scale computing resources and highly accurate numerical discretizations to capture the scale interactions that govern the aerothermal behavior of the targeted problem. Indeed, the multiscale nature of these simulations requires very huge isotropic meshes to capture the small scales. The classical use in the literature is to build a static huge mesh at the beginning of the simulation. On the contrary, the approach of the ARTEMIS project focuses on the design of energy-aware algorithms which are dynamic and adaptive to the multi-scale solutions.

However, as mentioned before, combining these mesh adaptation techniques with unsteady Navier Stokes solvers remains a challenge in the literature. Moreover, designing an efficient parallel adaptive mesh and solver tools is still also a tough HPC challenge. In this project, we try to meet these challenges and bring some answers in the context of aerothermal simulations. Therefore, we propose in this project to launch massively parallel aerothermal computations using Cimlib-CFD on the GENCI Occigen II supercomputer in order to assess for the parallel capabilities of our code.

5.3.2.2 Presentation of the Occigen supercomputer

The GENCI Occigen supercomputer counts 34 racks where 27 racks are dedicated to computation and 7 racks ensure connections, user services and data management. The cluster is split into two different slices called Haswell and Broadwell:

- Haswell counts 50 544 cores distributed over 2 106 nodes of 2 processors Intel with 12 cores each. The processors Intel 12-Core E5-2690 V3 allows a nominal computation power of 124 GigaFLOPS.

- Broadwell counts 35 280 cores distributed over 1 260 nodes of 2 processors Intel with 14 cores each. The processors Intel 14-Core E5-2690 V4 allows a nominal computation power of 145 GigaFLOPS.

These characteristics are summarized in Table 5.2. Haswell is split into two parts. Half of the nodes have a memory capacity of 64 Go whereas the other half has a memory capacity of 128 Go. Within these two different memory capacities, between 6 and 13 Go are used for system operations. The connection network is also an infiniband network.

Name	Processor Family	# of cores	Core frequency (Ghz)	Cache Memory (MB)	Memory (GB)
Haswell (part 1)	Xeon-E5-2690V3	24	2.6	30	64
Haswell (part 2)	Xeon-E5-2690V3	24	2.6	30	128
Broadwell	Xeon-E5-2690V4	28	2.6	35	64

Table 5.2 – Characteristics of the SMP nodes on the GENCI Occigen II supercomputer

Concerning job management, the GENCI Occigen II supercomputer uses the Simple Linux Utility for Resource Management (SLURM) workload manager introduced by Yoo et al. in [114]. It has globally the same functions than the OAR environment. First, it allocates exclusive and/or non-exclusive access to resources to users for some duration of time so they can perform work. Then, it provides a framework for starting, executing, and monitoring work on the set of allocated nodes. Finally, it arbitrates contention for resources by managing a queue of pending work.

For scheduling reason and in order to handle as best as possible the queue of pending works, the GENCI Occigen II supercomputer does not allow jobs with a wall-time superior to 24 hours. In our case, it causes a major problem because most of our unsteady aerothermal simulations require a total running time of more than 24 hours, even by using the large computing resources of the supercomputer. Therefore, it was necessary for us to implement a checkpoint-restart process, using the SLURM environment, in order to allow for our simulations to exceed 24 hours of computational time. This process is described in the next section.

5.3.2.3 Checkpoint-restart process

The checkpoint-restart process designed during this PhD allows launching successive jobs for one same simulation. It was implemented following a certain number of constrains:

- the simulation have to run from the start to the end without any intervention from the user,

- for the reasons explained in the previous section, each job require a running time inferior to 24 hours,

- if an hardware component fails, in order to be fault tolerant, the successive jobs have to be relaunched on the other working components,

- it is not possible to use a shell daemon to schedule the different jobs because the machine does not allow to have background daemons running for more than 30 minutes on the front nodes.

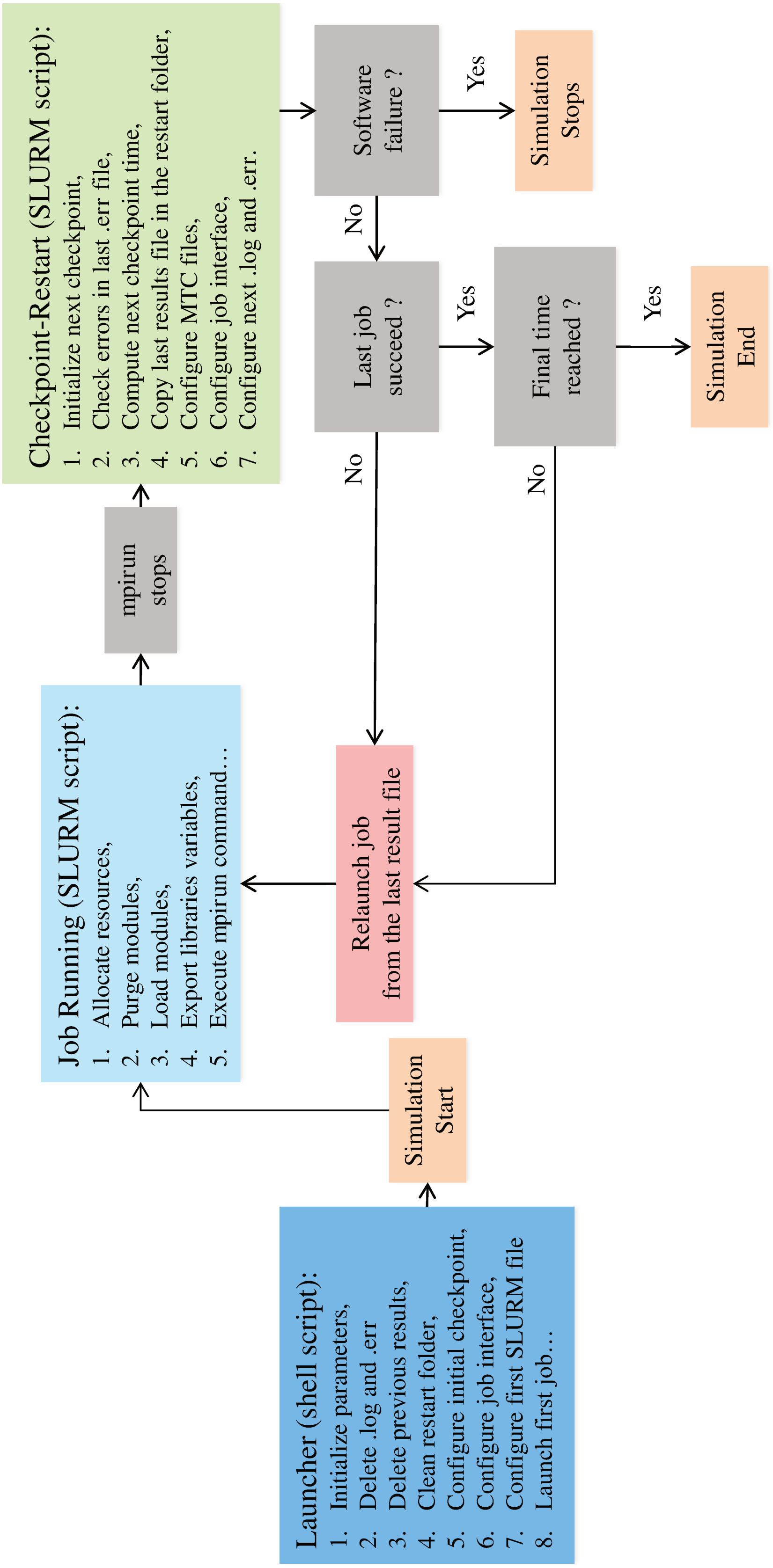

Launcher (shell script):
1. Initialize parameters,
2. Delete .log and .err
3. Delete previous results,
4. Clean restart folder,
5. Configure initial checkpoint,
6. Configure job interface,
7. Configure first SLURM file
8. Launch first job…
Simulation Start
Job Running (SLURM script):
1. Allocate resources,
2. Purge modules,
3. Load modules,
4. Export libraries variables,
5. Execute mpirun command…
mpirun stops
Checkpoint-Restart (SLURM script):
1. Initialize next checkpoint,
2. Check errors in last .err file,
3. Compute next checkpoint time,
4. Copy last results file in the restart folder,
5. Configure MTC files,
6. Configure job interface,
7. Configure next .log and .err.
Software failure ?
Yes
Simulation Stops
No
Last job succeed ?
Yes
Final time reached ?
Yes
Simulation End
No
No
Relaunch job from the last result file

Figure 5.5 – Checkpoint-restart process on the GENCI Occigen II

A schematic representation of the designed checkpoint-restart process is given in Figure 5.5. The Launcher is a shell script that needs to be executed only at the beginning of the simulation. It first (i) initializes the simulation parameters, (ii) cleans the result and restart folders and (iii) configures the initial job. Then, a SLURM script is executed and the initial job starts. At the end of the execution of the *mpirun* command, the Checkpoint-Restart script is executed. It first initializes the variable for the next checkpoint. Then, it reads the error file (*.err*) in order to check if hardware component errors have appeared. Then, the next checkpoint time of the next job is updated allowing the simulation to keep advancing in time. Also, the last result file from the previous job is copied in the restart folder in order to keep the data from the restarting increment. Finally, the MTC files (piloting the software), the job interface and the next *.log* and *.err* files are updated in order to start the new job.

Depending on the errors, the simulation either continues or stops. Indeed, if the job has not succeeded but the error is an hardware error, independent from a software failure, a new job is launched from the last result file on the other working components and the simulation continues. This way, our procedure is fault tolerant in case a failure from an hardware component occurs. On the contrary, if the error is due to a software problem (like memory leak, segment fault or others...), the simulation crashes and the checkpoint-restart process stops automatically.

The proposed checkpoint-restart process has been tested and validated on the GENCI Occigen II supercomputer. It works well and with this process, the simulation can reach the end even if hardware errors occurs. However, the main drawback of this process is that it necessitates from the user to know approximately how much increments will be computed in 24 hours. In fact, the user has to acknowledge an intermediate number of increment for the successive jobs. In particular, if no hardware error occurs, the *mpirun* command will stops only if the intermediate increment is reached and it is only then that the checkpoint-restart process will be able to start.

5.4 Applications

In this section, we present different applications that illustrate the parallel computational framework introduced in this chapter. The first application concerns the well known benchmark of the 3D ventilated cavity studied in Chapter 3 and we propose a strong scalability analysis on the two solvers used during this PhD work. The second application is the single impingement jet cooling studied in Chapter 2. On the latter, we propose an application of the parallel mesh adaptation procedure and, in particular, we apply this procedure on the two hardware systems previously introduced. Finally, the last application concerns the industrial application of a complete turbine vane and aims at demonstrating the parallel capabilities of Cimlib-CFD on the GENCI Occigen II supercomputer.

5.4.1 3D ventilated cavity

This first application refers to the 3D ventilated cavity presented in Chapter 3. For this case, we propose a strong scalability analysis. Therefore, we use the fixed mesh presented in the former chapter but we realize four computations over 28, 56, 84 and 112 processors. Indeed, these computations are launched on the Cluster Intel over 1, 2, 3, or 4 nodes of 28 processors. Doing so, as required by the strong scalability analysis, we keep a fixed problem size but we vary the number of processors. The obtained partitions are given in Figure 5.6. In this figure, we observe that the partitions respect our expectations in terms of distribution on the geometry. In particular, for the last computation, the interfaces are well minimized.

The computational time of each computation is given in Table 5.3. For this case, we focus the analysis on the two linear systems solved during the simulation, i.e. the Navier-Stokes VMS solver

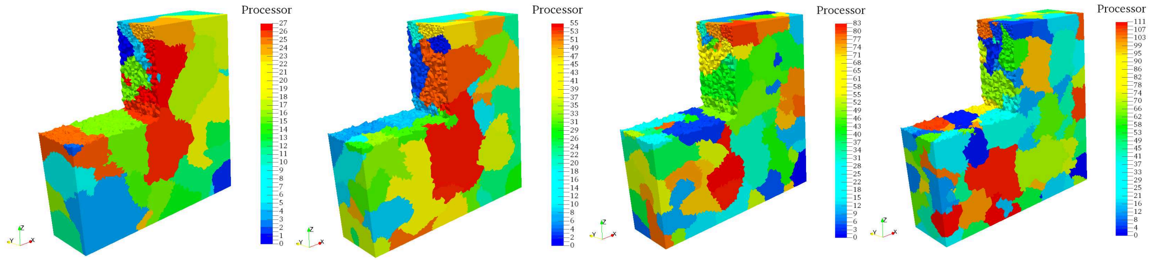


Figure 5.6 – Mesh partitioning over 28, 56, 84 and 112 processors

and the convection-diffusion solver. The computational time data are extracted over 1000 increments, from the beginning of the permanent regime. Indeed, the results presented in the table are the mean computational time required to solve each system, averaged over 1000 increments. The first observable result is that the Navier-Stokes VMS solver requires more computational time than the convection-diffusion solver. This is due to the fact that, for each point of the mesh, the Navier-Stokes solver has 4 DOF (v_x, v_y, v_z, p) to solve, instead of 1 DOF (T) for the thermal convection-diffusion solver.

Name	Nb processors	Navier-Stokes VMS	Speed-up	Convection-Diffusion	Speed-up
Computation 1	28	17.93	1	0.232	1
Computation 2	56	8.39	2.14	0.102	2.28
Computation 3	84	5.16	3.47	0.060	3.86
Computation 4	112	3.61	4.97	0.043	5.40

Table 5.3 – Mean computational time (in seconds) and speed-up on the Cluster Intel

Along this table, we present in Figure 5.7, the scalability charts of the two solvers. In particular, we plot the speed-up realized by doubling, tripling and quadrupling the allocated resources. The time reference to compute these speed-ups is the computational time corresponding to one node, i.e. to 28 processors. These speed-up charts both show linear behaviors with a determination coefficient close to one. This information give us confidence on the predictability and the scalability of the algorithm used to solve the two linear systems in Cimlib-CFD. We notice that the speed-up values exceeds the perfect scalability expectations. Indeed, a perfect strong scalability should give theoretical values of 2, 3 and 4 for the different computations. However, this analysis would not take into account the time's overhead due to communications. In particular, we notice that the communications between the processor represent a more important part of the resolution time when there are less DOF in the linear solver. For this reason, the speed-up of the convection-diffusion solver is depreciated in comparison with the one for the Navier-Stokes VMS solver. However, comparing the proportionality coefficient of these speed-up charts, we observe that the convection-diffusion solver show a better scalability behavior that the Navier-Stokes VMS solver.

5.4.2 Single impingement jet cooling

This second application allows testing the parallel mesh adaptation procedure presented in this chapter. In particular, we test it on the two hardware systems presented previously. To do so, we launch two distinct computations: *Computation 1* on the Cluster Intel and *Computation 2* on the GENCI Occigen II supercomputer. The details of these two computations are given in Table 5.4. Referring to Chapter 4, in order to follow the flow dynamics during the unsteady simulation, the parallel mesh adaptation procedure is done every 10 increments over the 5 000 increments. Therefore, for this simulation, 500 remeshing, repartitioning and load balancing procedures are

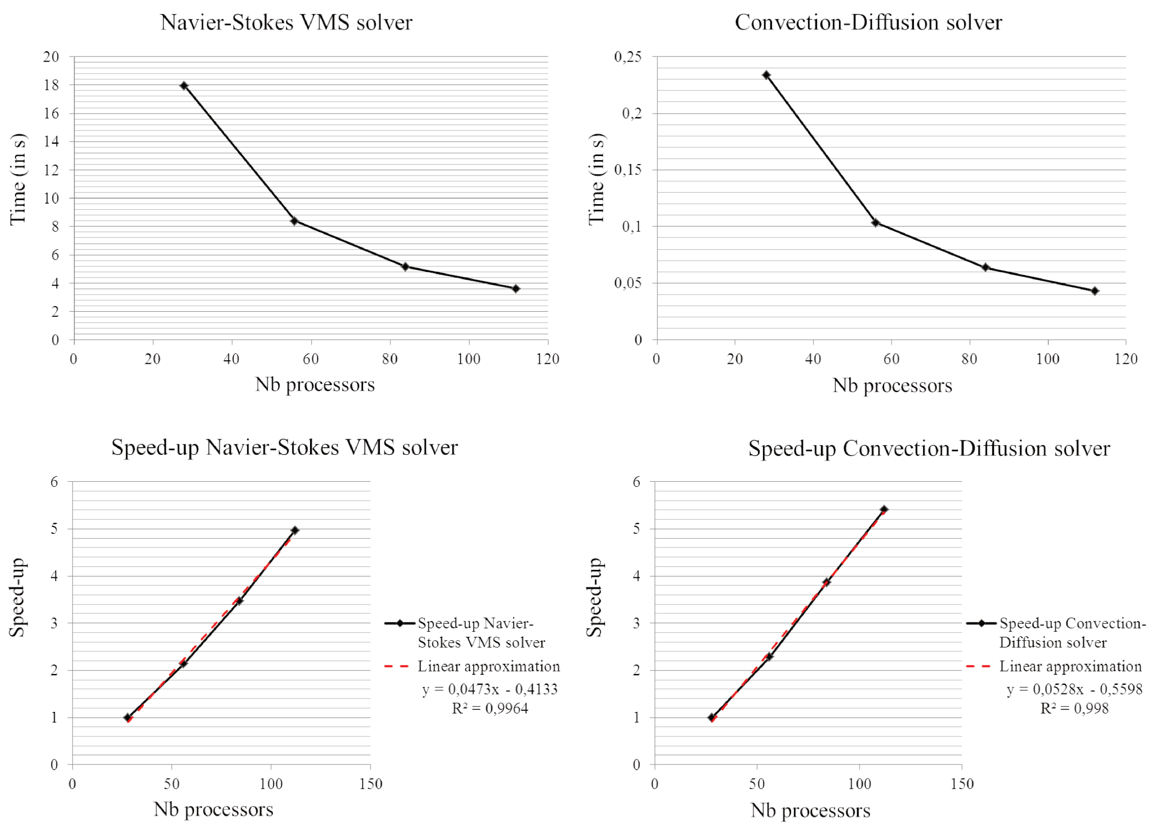


Figure 5.7 – Strong scalability of the Navier-Stokes VMS solver and the Convection-Diffusion solver

realized. This quite large number of mesh adaptation steps shows the importance to have efficient remeshing, repartitioning and load balancing algorithms in order to reduce the computational costs of parallel mesh adaptation.

Name	System	Init. Nb Elements	Targeted Nb Elements	Allocated Cores
Computation 1	Cluster Intel	800 000	4 000 000	28
Computation 2	Occigen II	6 000 000	12 000 0000	504

Table 5.4 – Description of the two computations for parallel mesh adaptation

Computation 1 is launched on a single Cluster Intel node of 28 cores. Its 3D mesh corresponds to the one presented in Chapter 4 Section 4.5. We start from an initial isotropic mesh of 800 000 P1 elements and a constrain of 4 000 000 P1 elements is imposed to the mesh adaptation procedure. The error estimate used here is $\eta_{\Omega_e,new}$ corresponding to the new multiscale mesh adaptation method developed in Chapter 4. A sample of the obtained partitions using this dynamic mesh adaptation method is given at different increments in Figure 5.8. In this figure, we observe that the number of processors that share a part of the impact plate increases in time. Indeed, as we adapt the mesh using the temperature variations, the number of elements increases on the impact plate and thus, as expected, the number of processors allocated to this part of the mesh also increases.

Computation 2 is launched on 21 nodes of 24 cores on the GENCI Occigen II supercomputer. The checkpoint restart procedure is applied every 200 increments. We start from an initial isotropic mesh of 6 000 000 P1 elements and a constrain of 12 000 000 P1 elements is imposed for the mesh adaptation procedure. As for the previous case, the resulting partitions are given in Figure 5.9. In this figure, we observe that the mesh is repartionned, as expected, following the cells distribution

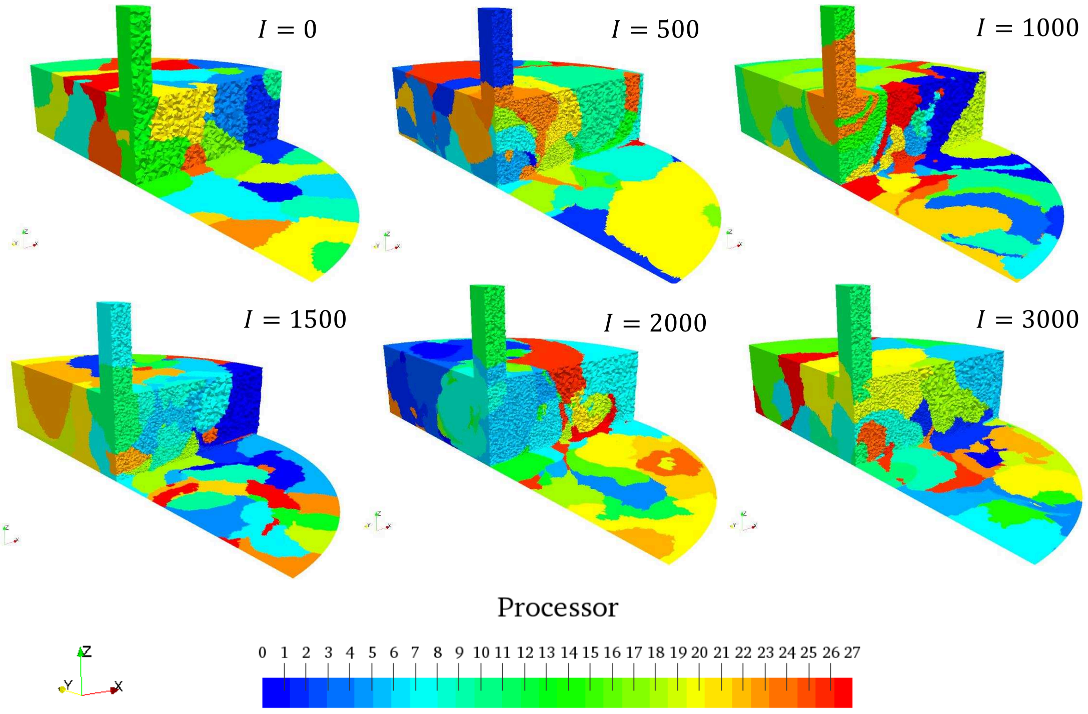


Figure 5.8 – Dynamic mesh partitioning over 28 processors: partition at different increments

presented in Chapter 4. Indeed, if we look closely at the plate, the processors distribution seems to follow closely the mesh density in the area of interest. In fact, the mesh partitioning algorithm follows dynamically the capture of the turbulent structures on the plate and allocates a larger number of resources where these structures are developing.

Finally, these two computations show our ability to launch the parallel mesh adaptation procedure on two different systems and at two different scales.

5.4.3 Turbine vane cooling

The close proximity between our research work and the industrial world represented by SAE continuously obliges us to deal with more and more complex simulations. The present study aims at characterizing the flow field in a schematic but realistic turbine vane. In fact, this work aims at simulating the turbine vane cooling induced by 39 holes in interaction with themselves in a complex geometry. In the literature, numerical and experimental references of the investigated configuration can be found in the research works of Fenot et al. and Laroche et al. in [115] and [116] respectively.

5.4.3.1 Geometry and time discretization

The test section of the geometry is presented in Figure 5.10. In this figure, we can see that the air is injected through a central cavity section of length 12 mm and of height 20 mm. Then, the air goes into the 39 holes distributed in 3 rows of 5 holes on the extrados, 1 row of 9 holes at the leading edge and 3 rows of 5 holes on the intrados. The central cavity is a closed space except for the inlet section and the trailing edges of the intrados and the extrados. Therefore, the air is inevitably evacuated through the trailing edges of the intrados and the extrados.

From this test section, SAE generated a fluid geometry given in Figure 5.11. Indeed, in this study, we do not need to consider the material characteristics of the turbine vane itself. Therefore, only the fluid is considered. The entry of the fluid into the domain is realized along the Z axis,

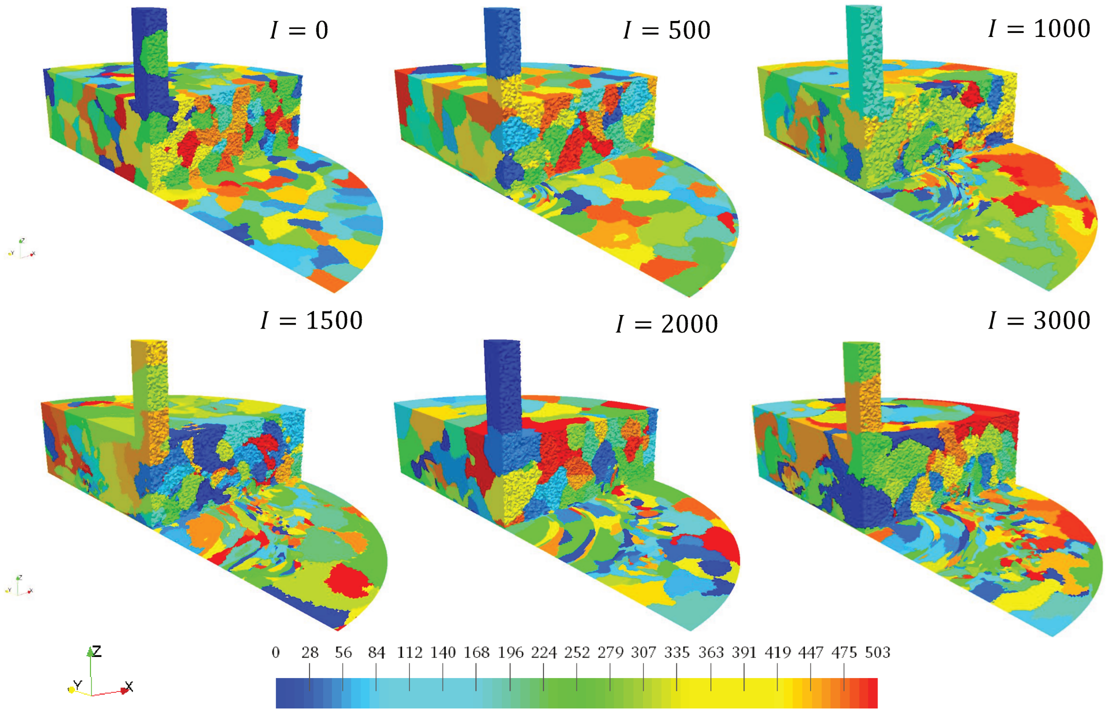


Figure 5.9 – Dynamic mesh partitioning over 504 processors: partition at different increments

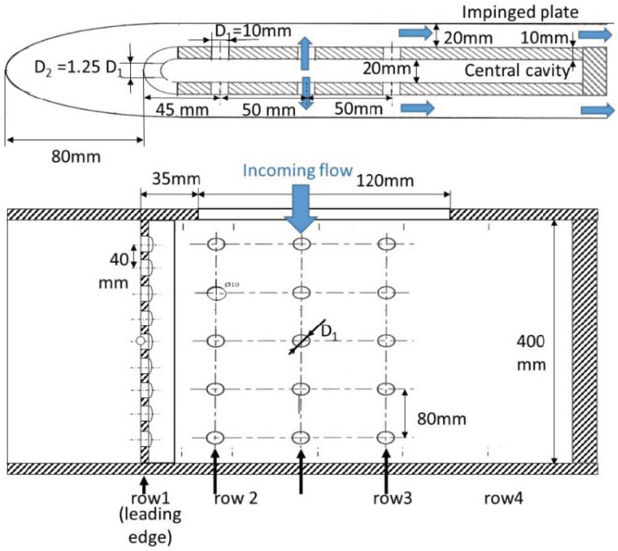


Figure 5.10 – Test section of the turbine vane [116]

through a convergent rectangular nozzle, visible in Figure 5.11. Then, the fluid heterogeneously feeds the 39 injection holes and impacts the leading edge and the intrados/extrados planes. Finally, the flow exits the geometry along the X axis through the trailing edges of the intrados and extrados.

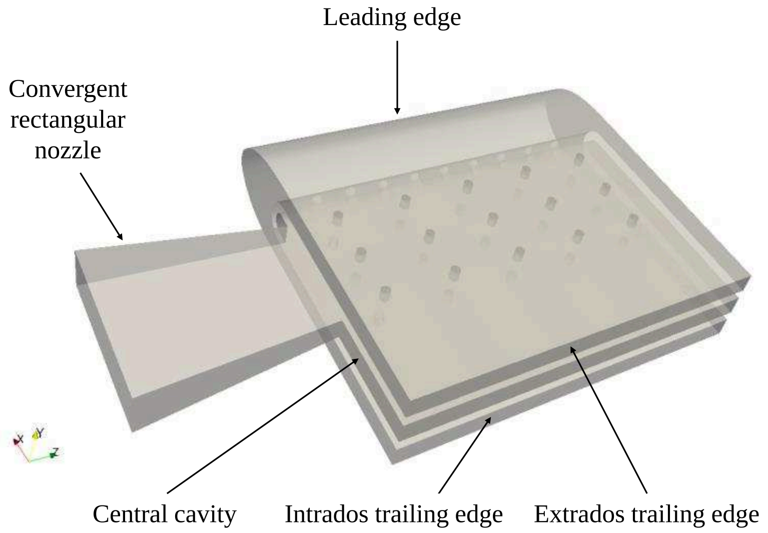


Figure 5.11 – Fluid geometry of the turbine vane

5.4.3.2 Numerical models and boundary conditions

In this work, we propose to compute velocity and the pressure fields $(\mathbf{v}, p)$ using the implicit iterative VMS solver for the unsteady incompressible Navier-Stokes equations. Concerning the aerodynamic boundary conditions, an input velocity corresponding to the mass flow rate (60 g/s) prescribed by the experimental study of Fenot et al. in [115] is imposed at the convergent nozzle entry section. Then, an atmospheric static pressure and output velocities are imposed on the intrados/extrados trailing edges sections. These output velocities are chosen with respect to the mass flow rate equilibrium inside the overall geometry. In the middle part, corresponding to the central cavity, the output section is a solid wall with a null Dirichlet boundary condition. For the reader understanding, a representation of these boundary conditions is given in Figure 5.12. For all the other borders of the domain, null Dirichlet boundary conditions are imposed. At the beginning of the simulation, the velocity is null everywhere except on the boundary conditions.

To solve the partial differential equation associated to the thermal problem, we use a stabilized finite element method defined by the SUPG scheme introduced in Chapter 3. Concerning the thermal boundary conditions, as recommended in [116], the initial temperature and the air temperature T_{ref}, injected into the convergent nozzle is equal to 296K. Then, we consider mainly two impinged plate corresponding, in the Y axis direction, to the top (extrados) and the lower (intrados) part of the geometry. On these two plates a constant heat flux $\Phi_{conv} = 1600\mathrm{W}/m^2$ is imposed. In this study, we look for the local convective heat transfer coefficient h_{conv} on the intrados/extrados planes defined by the following formula:

$$h_{conv} = \frac{\Phi_{conv}}{\mathrm{T}_{wall} - \mathrm{T}_{ref}} \tag{5.3}$$

with T_{wall} the wall temperature. Therefore, the resulting variable of interest is T_{wall} on the two impinging plates.

The two additional numerical models implemented in Chapter 3 to ensure a strong aerothermal coupling are also used here. Indeed, the modified dynamic viscosity μ_{total} is used as an entry in the Navier-Stokes VMS solver; and the modified thermal conductivity λ_{total} is used as an entry in the convection-diffusion SUPG solver.

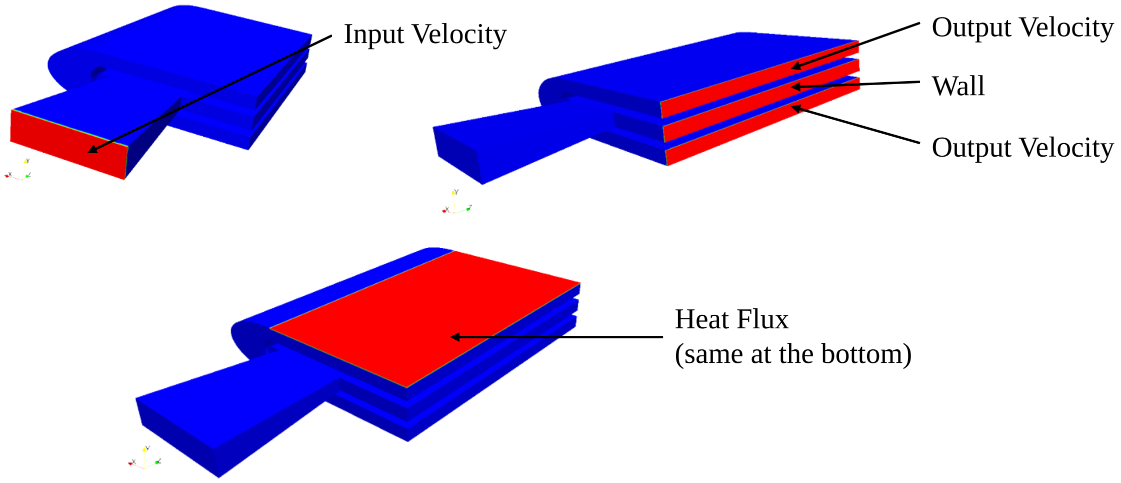


Figure 5.12 – Illustration of the boundary conditions of the turbine vane

Finally, the time discretization is chosen by taking into consideration the velocity boundary conditions and the size of the geometry. After some experiments on the geometry, the simulation time step is taken as $\Delta t = 0.01$ s and the total simulation duration is taken as $t_{total} = 50$ s giving a total of 5 000 simulation increments.

5.4.3.3 Incremental space discretization

From the fluid geometry, we generate unstructured isotropic tetrahedral 3D meshes. In order to test our parallel numerical tools, we propose in this work, an incremental space discretization. In fact, 3 meshes M1, M2 and M3 counting respectively 500 000, 7 000 000 and 25 000 000 elements are generated. These meshes are presented in Figure 5.13. In this figure, we observe that, in order to obtain a homogeneous distribution of the cells inside the geometry, at least 25 millions elements are necessary. Concerning M3, if we look inside the holes, we can count around 15 cells along the hole's diameter corresponding to the coarse mesh of Laroche et al. in [116].

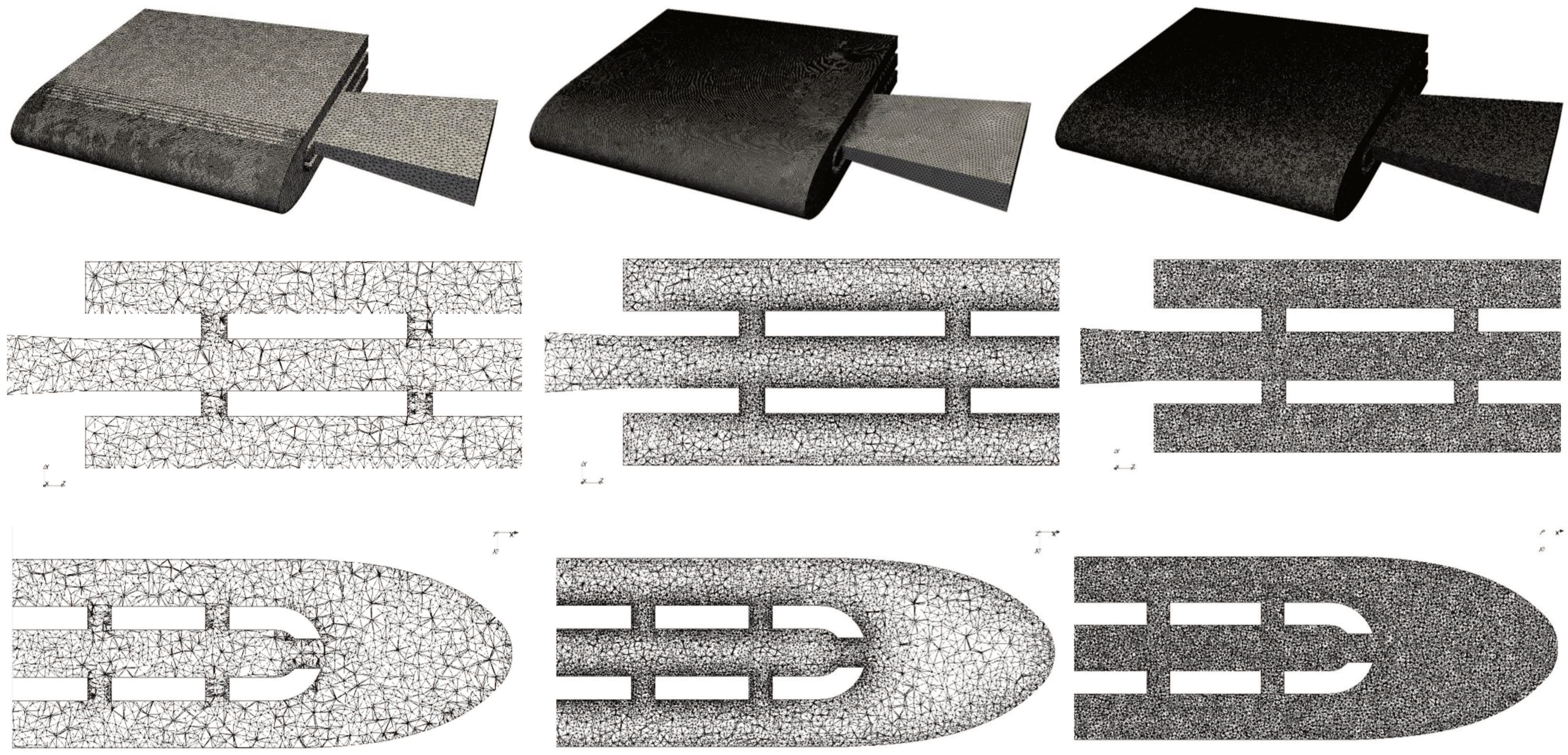


Figure 5.13 – Unstructured isotropic tetrahedral 3D meshes counting (from left to right): M1: 500 000, M2: 7 000 000 and M3: 25 000 000 elements; (top) Global view; (middle) Slice view in plane YZ at $x = -45\ mm$; (bottom) Slice view in plane XY at $z = -0\ mm$

5.4.3.4 Allocated resources

The three computations corresponding to the three meshes previously described are launched on the GENCI Occigen II supercomputer. We adapt the requested resources to the size of the mesh. After numerical experiments, it was decided to allocated one CPU for 25 000 elements of the meshes. The allocated resources for each mesh are given in Table 5.5.

Mesh name	Nb elements	Nb cores/nodes	Nb nodes	Nb cores
M1	500 000	20	1	20
M2	7 000 000	24	12	288
M3	25 000 000	24	42	1008

Table 5.5 – Allocated resources for each case on the GENCI Occigen II supercomputer

As we do not apply parallel mesh adaptation in these cases, the mesh is partitioned only at the beginning of the simulation. The partitioned meshes are presented on Figure 5.14. In this figure, we observe that the different processors are homogeneously distributed over the meshes, even if the geometry is relatively complex. In particular, the partitions' interfaces are well minimized.

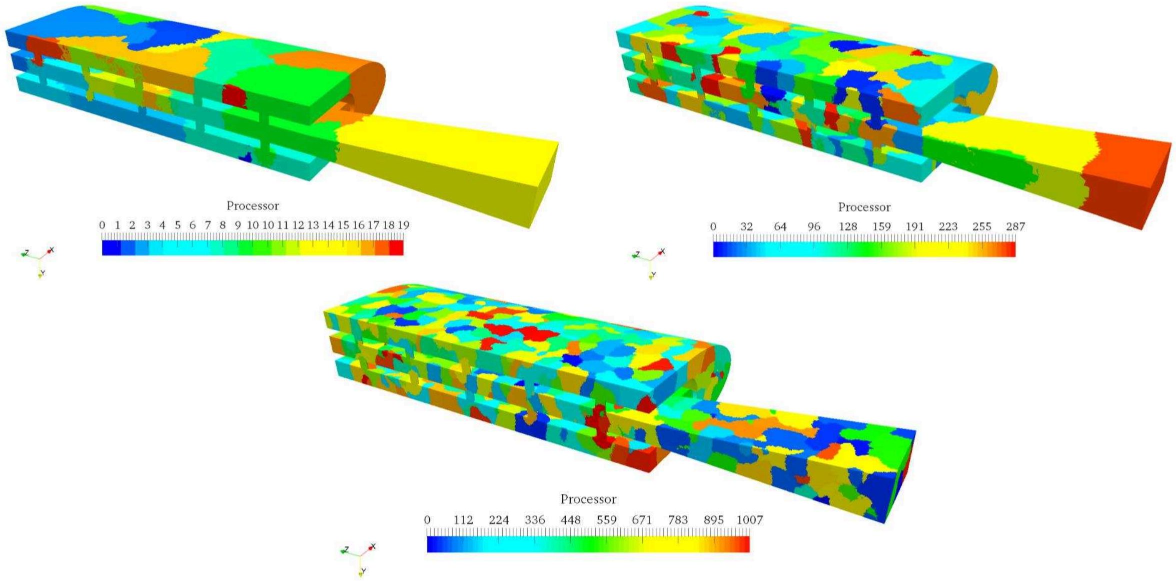


Figure 5.14 – Partitioned meshes of the turbine vane for: (top left) M1: 500 000 elements on 20 cores, (top right) M2: 7 000 000 on 288 cores and (bottom) M3: 25 000 000 elements on 1008 cores

For these simulations, the checkpoint-restart procedure needs to be applied because their total computational times exceed 24 hours. After some numerical experiments on the supercomputer, it was decided to put a checkpoint-restart procedure every 400 increments. Therefore, 13 checkpoint-restart procedures were needed to reach the end of the simulation.

5.4.3.5 Computational time

In this section, we propose an analysis of the weak scalability of our finite element code. Indeed, the allocated resources are adapted to the three computations allowing to keep a constant number of elements per cores. The data concerning the computational time of each of the three computations are presented in Table 5.6. This computational time is given over 100 increments, 1000 increments and over the total simulation duration. It has to be noted that this computational time takes into account the total computation time of our code. In particular, it takes into account:

- the initial mesh partitioning time,
- the execution times of the Cimlib-CFD models (solvers etc.),
- the writing time of the results on the disk,
- the execution of the checkpoint-restart procedure,
- the queuing time of the successive jobs,
- the reading and repartitioning time of the last result file.

Taking this global metric to measure the computational time does not allow to do a fair scalability analysis of our finite element code. In fact, too many variables, like the queuing time of the successive jobs, are out of our control reach. However, it has the advantage to show, at a global and practical scale, the behavior of Cimlib-CFD on the massively parallel supercomputer. In particular, looking at the results concerning M3, we observe that the total computational time for the simulation represents 277 704 CPU hours which is expensive in comparison to the other cases presented in the PhD work.

Mesh name	100 increments	1000 increments	Total
M1	1 h 02 min	0 days 14 h 49 min	2 days 12 h 00 min
M2	1 h 55 min	1 days 04 h 33 min	4 days 05 h 28 min
M3	4 h 11 min	2 days 01 h 19 min	11 days 11 h 31 min

Table 5.6 – Computational times on the GENCI Occigen II supercomputer

Along Table 5.6, we draw in Figure 5.15 the charts of the computational times on the GENCI Occigen II supercomputer in function of the number of elements for the three meshes M1, M2 and M3. These charts are given over the different ranges of increments. They allow to assess for the weak scalability performances of Cimlib-CFD on the supercomputer. Keeping a constant number of element, a perfect scalability should show the same computational times for the three computations. However, this analysis does not work if we consider the additional overhead times.

On the first chart, corresponding to the computational time over 100 increments, a linear behavior is observed. Indeed, the determination coefficient of the linear approximation is equals to 0.9997 which is a proof of a proper linear behavior. This shows that the overhead time behave as expected and varies linearly with the number of elements. When we consider the computational times exceeding 400 increments, a linear behavior is still observed but the determination coefficient is depreciated. This is caused by the effect of the checkpoint restart procedure and the successive jobs' queuing times previously mentioned. However, looking at the total computational times and the fairly good linear behavior of the chart, we can say that our finite element library shows good scalability performances on the machine.

5.4.3.6 Results and discussions

In this section we present the results in terms of aerothermal coupling parameters and velocity fields obtained on the three meshes previously presented. An analysis of these results is proposed along with some perspectives for future works.

5.4.3.6.1 Resulting aerothermal coupling parameters In the turbine vane cooling simulation, the hypothesis of highly convective heat transfer in turbulent flows is still valid. Indeed, referring to previous studies [116], the Reynolds number inside the holes can be estimated like follows:

$$Re = \frac{V_b D}{\nu_{fluid}} \approx \frac{22 \times 0.01}{15e^{-6}} \approx 14\,000 \tag{5.4}$$

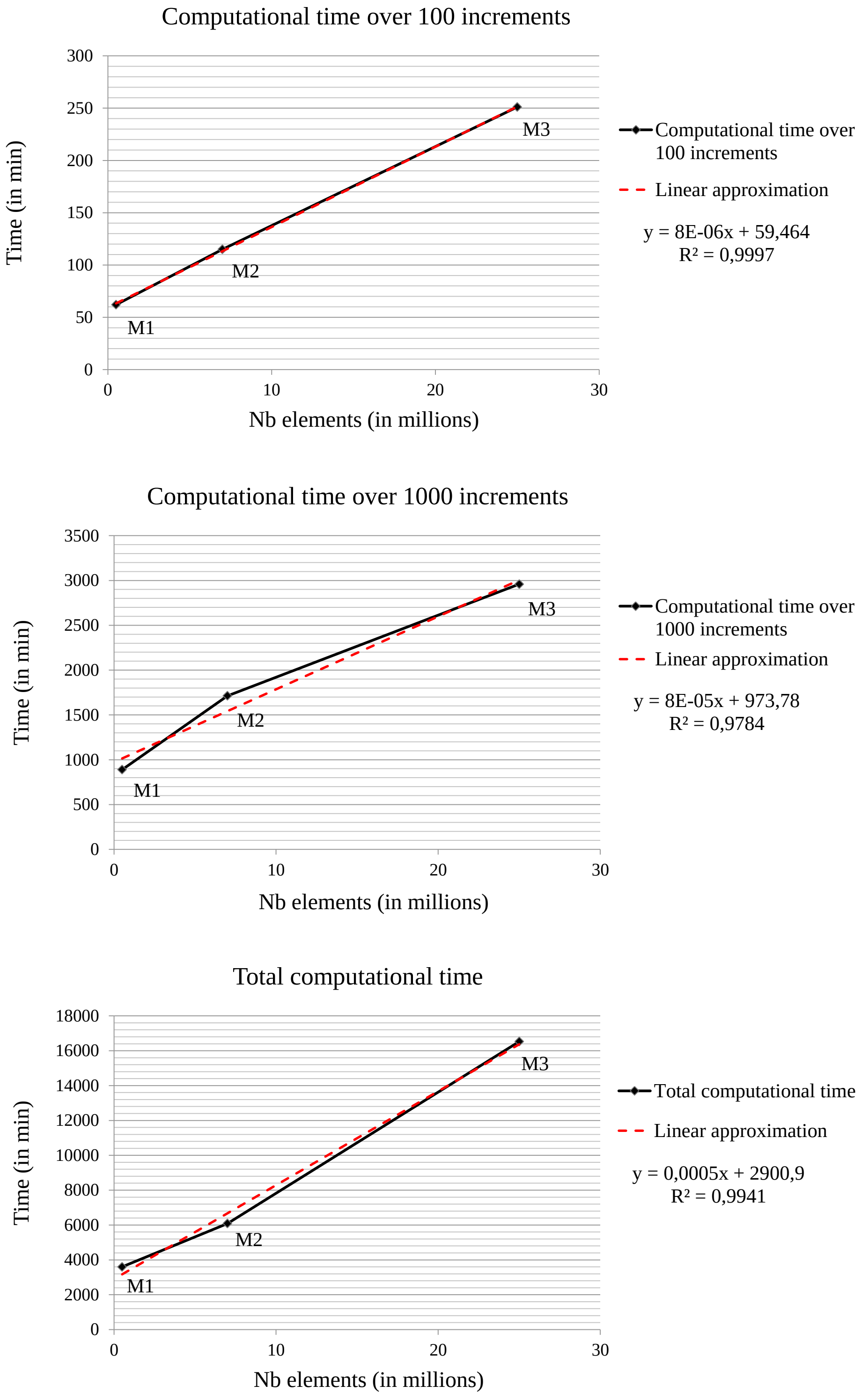


Figure 5.15 – Computational times for M1, M2 and M3 over diverse increment ranges

Therefore, the hypothesis of highly convective heat transfer in a turbulent flows is still valid. For this reason, we propose to use the two additional numerical models implemented in Chapter 3 to couple the Navier-Stokes VMS solver and the thermal convection-diffusion solver. We refer to this chapter for the theoretical aspects of the aerothermal coupling parameters computation. In particular, we propose here to use the Sutherland law and the Smagorinsky model to compute the modified dynamic viscosity μ_{total}. Doing so, the chosen aerothermal coupling configuration for the turbine vane cooling corresponds to *Computation B* of Table 3.2, which has shown to be the best choice in previous cases.

A snapshot of the resulting modified dynamic viscosity μ_{total} is given in Figure 5.16. In this figure, we observe that the turbulent dynamic viscosity is increased mainly in the holes where important velocity gradients are expected. Also, the variations of this modified dynamic viscosity is also important at the outlet of the geometry due to the imposed boundary condition. The use of a sub-grid scale turbulence model in these area has a smoothing effect on the solution and helps the convergence of the Navier-Stokes VMS solver.

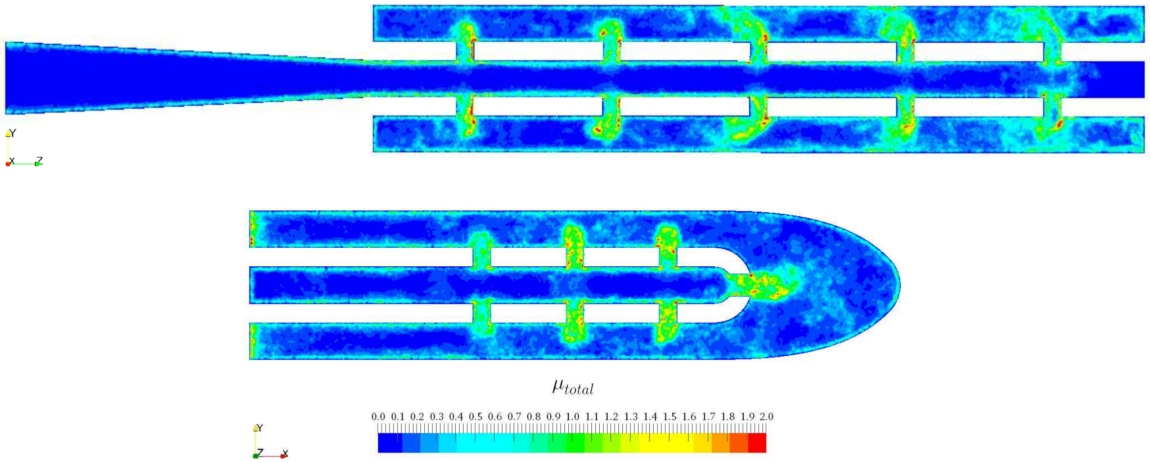


Figure 5.16 – Snapshot of the modified dynamic viscosity μ_{total} in plan YZ at $x = -45mm$ and in plan XY at $z = 0mm$

5.4.3.6.2 Resulting velocity fields The resulting instantaneous velocity field inside the turbine vane is presented in Figure 5.17. In this figure, we observe the flow, coming from the rectangular convergent nozzle, feeding progressively the 39 holes. In particular, we observe on each picture the flow acceleration caused by the convergence of the nozzle. The first columns of holes are rapidly fed but we need to wait until increment I = 1250 to see the flow going through all the holes.

From this point, the flow motion inside the central cavity continues to evolve until the permanent regime is reached (around increment I = 5000). Following this evolution, we clearly see the formation of a main vortex structure creating a massive flow re-circulation inside the central cavity. This re-circulation seems to find its origins at the crossing of the rectangular nozzle with the central cavity, but it also seems to be caused by the closed wall on the other side of the cavity.

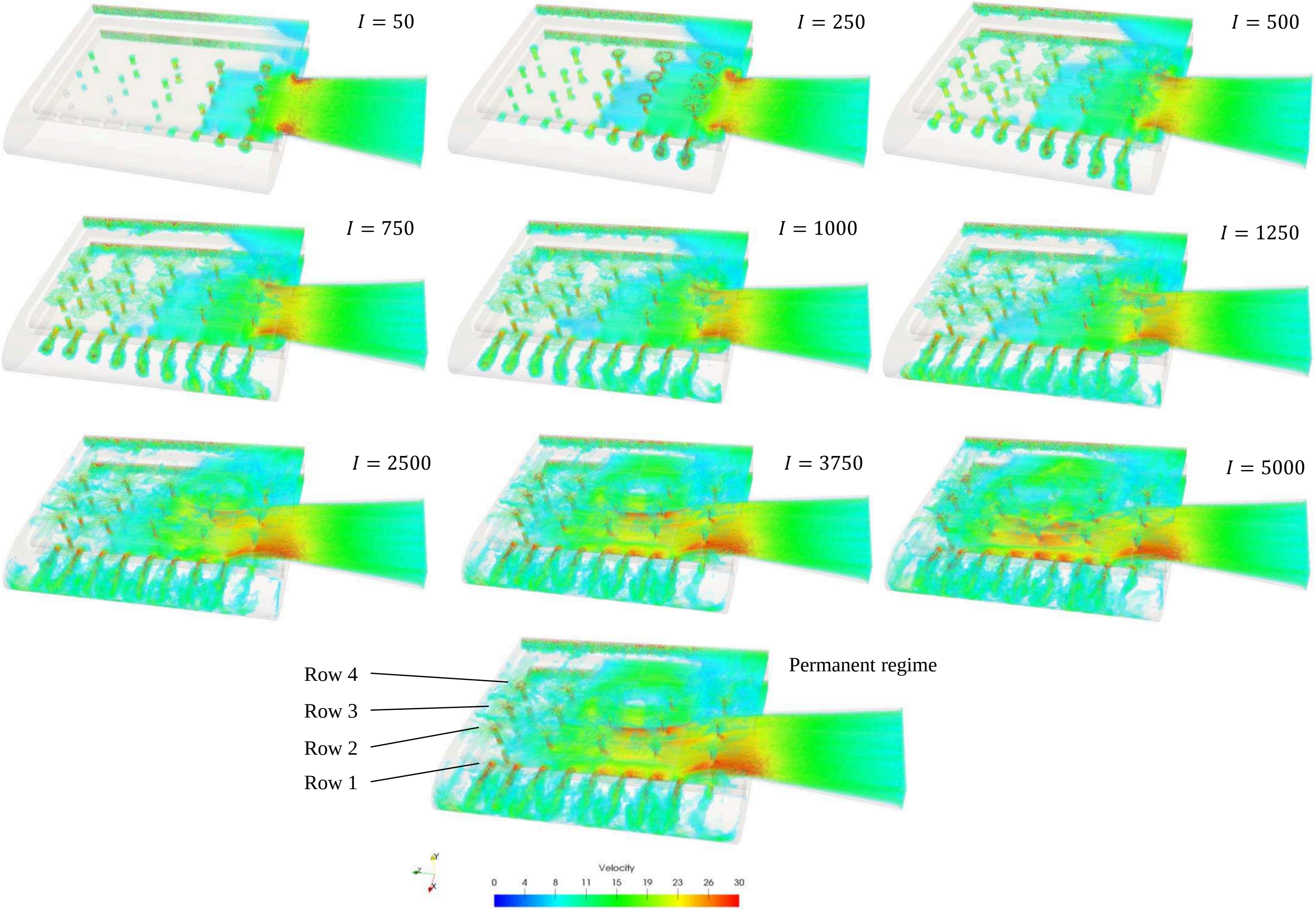


Figure 5.17 – Resulting instantaneous velocity field inside the turbine vane

We observe that this re-circulation has an important effect on the feeding of the different holes. Indeed, it creates a heterogeneous distribution of the mass flow rate between the holes. This important result was also pointed out in the work of Laroche et al. in [116]. In this paper, the authors measured the mass flow rate going through each hole and important differences were found from a hole to the other. Looking at their results presented in Figure 5.18, this heterogeneity in the mass flow rate can clearly be explained by the presence of the re-circulation, highlighted by our results. In fact, the hole at the center of the re-circulation is less fed due to the local slow down of the velocity at this particular location. From Figure 5.18, we can note that this slow down is located on row 4. In our results in Figure 5.17, we can also see that the center of the re-circulation is indeed located on row 4 and this gives us confidence on the validity of our computation.

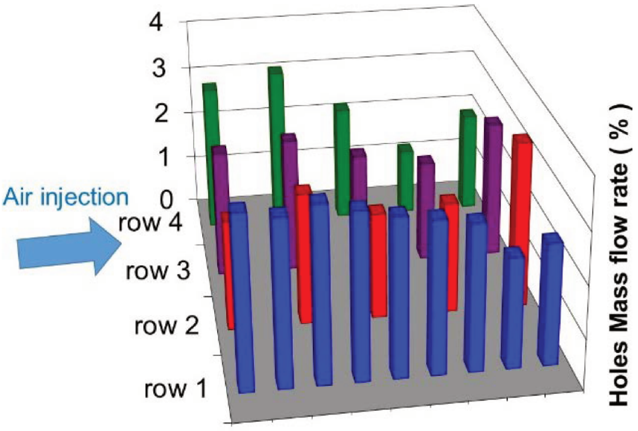


Figure 5.18 – Mass flow rate inside the holes

Furthermore, the resulting velocity fields in slice views are given in Figure 5.19 and Figure 5.20. We follow the work of Laroche et al. in [116] in order to compare our results with experimental and numerical references. Indeed, in the former work, the authors propose visualizations of the velocity field onto two planes: YZ at $x = -45mm$ and XY at $z = 0mm$. In particular, for the second plane, Particle Image Velocimetry (PIV) measurements coming from previous experimental studies are available. Thus, on this particular plane, we can compare our results with both (i) experimental PIV measurements and (ii) a numerical reference.

The results given in Figure 5.19 correspond to the slice in the YZ plane going across row 2. In this figure, we compare our results with the numerical ones from Laroche et al. in [116]. We observe that, for the three computations on M1, M2 and M3, the results fit globally well with the reference. Indeed, the velocity field inside the holes is well resolved by our simulations. In particular, the velocity magnitude of the jets is well respected. Furthermore, we observe that, by increasing the size of the mesh, the result gets better. Indeed, the computation on M3 shows impinging jets reaching the extrados and intrados at the same position than the reference. At the impact point, we recover solutions comparable to the one presented in the paper. The main differences between our results and the reference reside in the fact that we solve an unsteady problem. Doing so, it allows us observing more details concerning the flow turbulence on mesh M3. Indeed, in the last picture of Figure 5.19, we recover vortex structures that look like the ones we had in the single impingement jet cooling simulation. Inside the central cavity, our results are also comparable to the reference. Indeed, we can clearly observe the flow slowing down after it passes the second column of holes. Again, the order of velocity magnitude is also well respected.

This analysis on the flow turbulence still holds in Figure 5.20. In fact, in this figure, our results fit well the experimental and numerical references. In particular, we recover the expected velocity

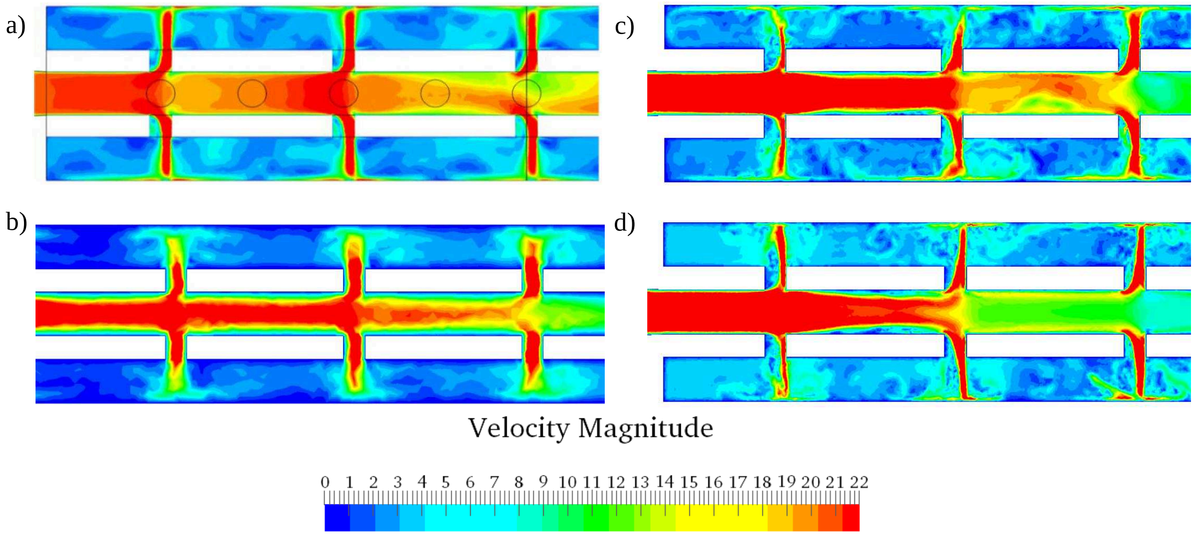


Figure 5.19 – Visualization of the averaged velocity field in plan YZ at $x = -45mm$ (i.e. the slice going through row 2). a) CEDRE $k-\omega$ SST from Laroche et al. in [116]; b) Results on M1; c) Results on M2; d) Results on M3

magnitudes in the flow. However, the fact that we use an unsteady simulation imply that our resolution of the flow turbulent behavior is better established. Indeed, in our results, we can observe the full flow turbulence of the jet arriving on the leading edge of the geometry. The complexity of our simulation is therefore increase but our understanding of the flow physic can be improved by these observations.

To finish, in order to illustrate our parallel mesh adaptation capabilities on such a complex industrial case, we apply the parallel mesh adaptation procedure presented in this chapter. This time, the criteria is based on the velocity field but we use the same metric construction that the one presented in Chapter 4. We start from the initial mesh M2, we wait for the pseudo-permanent regime to be established. Then, we activate anisotropic mesh adaptation with a targeted number of element of 7 000 000. This way, we only redistribute the elements inside the geometry, without increasing the size of the problem. The resulting mesh is presented in Figure 5.21. In this figure, we observe the possibilities that offers our parallel mesh adaptation framework. In fact, the mesh is able to follow the flow complexity inside the entire domain. In particular, we observe areas where the mesh follow the turbulent structures created by the rebound of the jets on the intrados and extrados. Furthermore, in this geometry, a quite important number of boundary layers appear and we can see that our mesh adaptation tool allows capturing several of them.

To conclude, we can say that, as we saw in Chapter 2 on the single impingement jet cooling and as pointed by the previous numerical studies, the accurate capture of highly convective heat transfer of impinging jets remains very challenging, and moreover in this kind of complex geometries. Indeed, we showed that it is first important to have a highly accurate solution of the velocity field inside the geometry, before even considering capturing the convective heat transfer. Our study brings an answer to this first difficulty but future works will have to focus more in details on the computation of the heat transfer.

In fact, the next steps of this work will be to obtain the convective heat transfer coefficient on the intrados, extrados and leading edge of the geometry. Achieving this is a real challenge. During this work, we tried to obtain this heat transfer but the meshes used were not sufficiently refined on the intrados and extrados area to hopefully obtain anything concerning them. In fact, if we link these results on 39 jets to the ones obtain in Chapter 2 on a single jet, we realize that we won't be

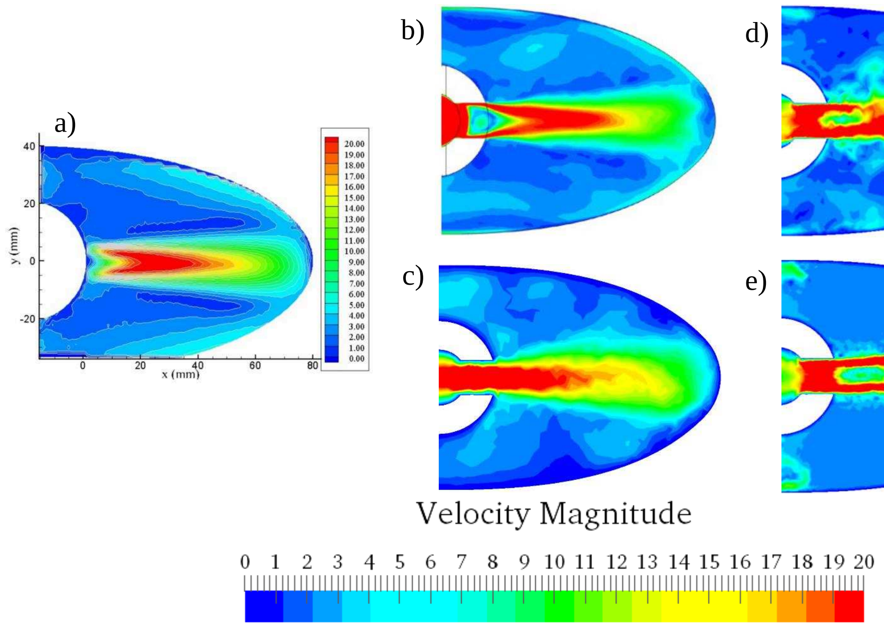


Figure 5.20 – Visualization of the averaged velocity field in plan XY at $z = 0mm$. a) PIV measurements; b) CEDRE ZDES from Laroche et al. in [116]; c) Results on M1; d) Results on M2; e) Results on M3

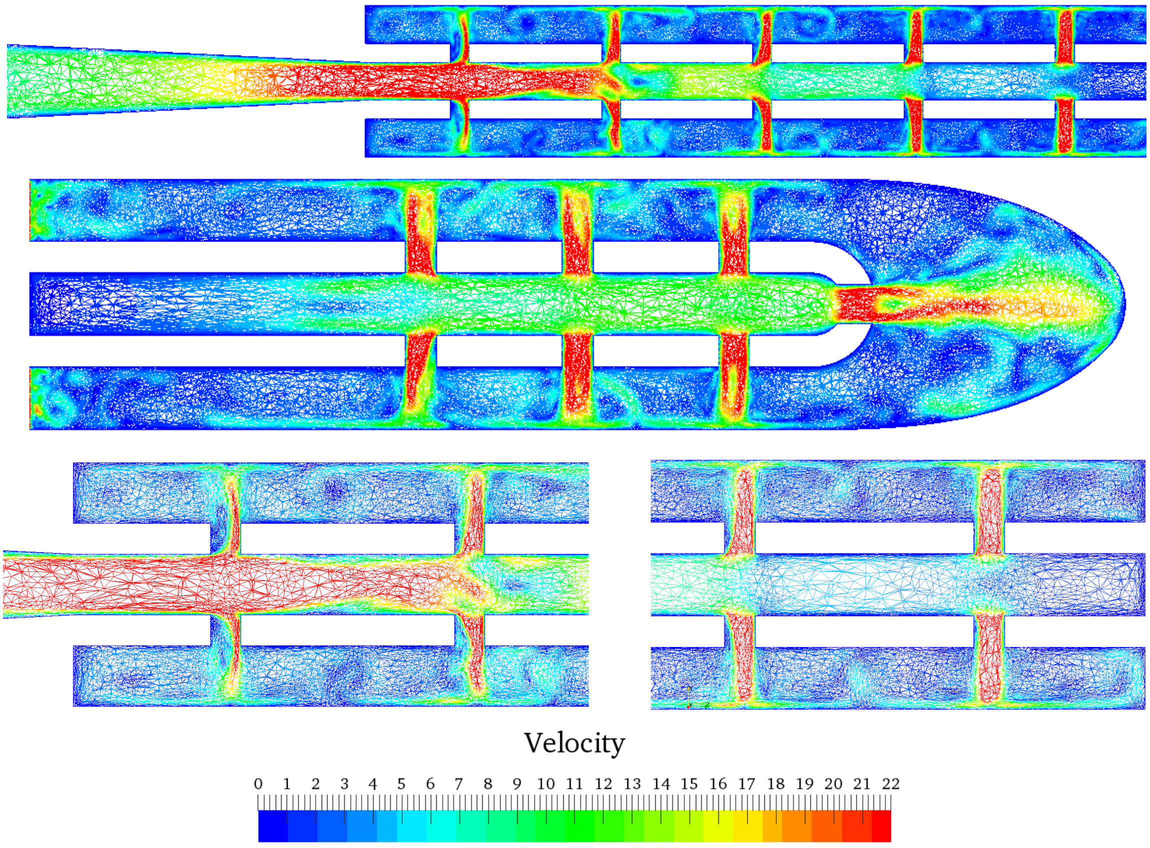


Figure 5.21 – Adapted mesh following of the complete turbine vane

able to capture this highly convective heat transfer using numerical simulation at this scale. This shows that, in order to obtain accurate results, we still need to increase the problem size and thus, the allocated computational resources. It shows the need to do computations at a larger scale and this work opens the way for future works on the turbine vane cooling with Cimlib-CFD.

5.5 Conclusion

To conclude, we proposed in this chapter a description of the parallel capabilities of Cimlib-CFD. First of all, we presented the FE framework, the mesh partitioning principles and the load balancing algorithms used in this PhD work. In particular, the parallel mesh adaptation procedure was presented and each step was detailed.

Second of all, we proposed a presentation of the two hardware systems used to develop numerical tools and to produce numerical results during the thesis. The first one was the lab's cluster, called Cluster Intel, allowing the development of numerical methods. The second one however, was the GENCI Occigen II supercomputer on which we executed Cimlib-CFD for the first time during this PhD work. This system allowed us to produce numerical results using massively parallel computations. However, the adaptation of Cimlib-CFD on this new system required, among others, the development of a checkpoint-restart procedure that we designed following a certain number of constrains imposed by the user policy and hardware restrictions of the supercomputer.

Then, we proposed application cases for parallel computations. The first one was dedicated to a strong scalability analysis of our finite element code on a well known benchmark. The results showed a linear increase of the speed-up with the allocated resources allowing us to conclude on the hard strong scalability of the numerical methods used to solve linear systems in Cimlib-CFD. Then, parallel mesh adaptation was tested on the two hardware systems for the case of the single impingement jet cooling. In this work, we showed the parallel capabilities of the software at two different scales by presenting dynamic parallel mesh adaptation over 28 cores on the Cluster Intel and then, over 504 cores on the GENCI Occigen II supercomputer. Our results showed that we obtained relevant partitions in both cases.

Finally, we presented in this chapter a more complex but realistic industrial case which concerned the cooling of a complete turbine vane composed by 39 holes. To study this geometry, an incremental space discretization was chosen along with adapted allocated resources allowing, in a certain way, to do a weak scalability analysis of our computations. The computational time on these incremental meshes showed a linear tendency which illustrated the good behavior of Cimlib-CFD on the supercomputer. Furthermore, the results showed our ability to obtain the unsteady turbulent velocity field inside this geometry but future works are still needed to obtain the heat transfer coefficients on the turbine vane. In particular, these future works will have to increase the size of the problem and to launch computations at an even larger scale.

Résumé du chapitre en français

Ce chapitre s'attache à décrire les techniques de parallélisation de calcul utilisées dans Cimlib-CFD. En effet, nous proposons d'abord une description détaillée des algorithmes de partitionnement et de répartition de la charge de calcul au niveau software. Également, la procédure d'adaptation parallèle de maillage est introduite et chacune des étapes clés de cette dernière est expliquée.

Nous proposons ensuite une présentation générale des deux systèmes de calcul utilisés lors de ce doctorat. Le premier est un système de calcul interne au laboratoire appelé Cluster Intel qui a plus particulièrement servi au développement des méthodes numériques introduites dans ce manuscrit. Le second système, en revanche, est un super-calculateur national appelé GENCI Occigen II sur lequel nous avons exécuté Cimlib-CFD pour la première fois pendant cette thèse. Ce super-calculateur nous a permit de produire des résultats numériques utilisant des calculs massivement parallèles. Cependant, l'adaptation de Cimlib-CFD à ce nouvel environnement de calcul a nécessité, entre autres, le développement d'une procédure d'arrêt et de reprise du calcul que nous avons conçu en respectant un certain nombre de contraintes imposées par la politique utilisateur du système.

Des cas d'application des techniques de parallélisation sont ensuite proposés. Le premier cas est dédié à l'étude de la scalabilité forte des solvers linéaires utilisés pendant la thèse et reprend le fameux cas de la cavité ventilée 3D. Les résultats montrent une augmentation linéaire du speed-up réalisé en augmentant les ressources de calcul. Cela nous permet de conclure sur la bonne scalabilité de nos méthodes de résolution de systèmes linéaires dans Cimlib-CFD. Ensuite, l'adaptation parallèle de maillage est testée sur les deux systèmes de calcul pour le cas du jet unique impactant. Dans ce travail, nous montrons les capacités parallèles du programme à deux échelles différentes, présentant d'abord une adaptation dynamique et parallèle du maillage sur 28 cœurs avec le Cluster Intel, puis sur 504 cœurs avec le super-calculateur GENCI Occigen II. Nos résultats montrent que l'on peut obtenir des partitions appropriées dans les deux cas.

Enfin, nous présentons dans ce chapitre un cas industriel plus réaliste qui concerne le refroidissement d'une aube de turbine complète composée de 39 trous. Pour étudier cette géométrie, nous proposons d'utiliser une discrétisation spatiale incrémentale en adaptant les ressources de calcul demandées à la taille du problème. Cela nous permet, dans une certaine mesure, de faire une analyse sur la scalabilité faible de notre code. Les temps de calcul sur ces maillages incrémentaux montrent une progression linéaire qui illustre le bon comportement de Cimlib-CFD sur le super-calculateur. De plus, les résultats montrent que nous sommes capable d'obtenir le champ de vitesse au sein de la géométrie. Cependant, des travaux futurs doivent être envisagés pour capture les coefficient de transferts thermiques. Pour cela, par comparaison avec les études numériques menées sur le cas du jet unique et plus largement, par les études numériques réalisées dans d'autres laboratoire, il sera nécessaire d'augmenter la taille du problème et de lancer des calculs à plus grande échelle.

Chapter 6

Conclusion and Perspectives

Contents

6.1 Conclusion

This PhD work was devoted to the development of a variational adaptive finite element method allowing to improve the aerothermal simulations related to the cooling of aircraft engines. More precisely, our goal was to develop a new multiscale mesh adaptation technique, well suited to the resolution of highly convective aerothermal problems encountered in particular in the turbine blade cooling. To study this complex aerothermal problems, it was necessary to focus our interest on a benchmark configuration of the single impinging jet cooling bringing into play highly convective heat transfer in turbulent flows.

The single impingement jet cooling problem was presented in Chapter 2. This chapter made it possible for us to better understand the complexity of the physical phenomenon occurring during the cooling. We proposed in this chapter two different numerical frameworks to solve this aerothermal problem. The first one used ANSYS CFX/Fluent and the second one used Cimlib-CFD. The preliminary study using the ANSYS framework clearly showed the need to have a fully 3D turbulent and unsteady resolution of the problem in order to capture the very specific heat transfer distribution of the impingement jet cooling on the impact plate. To do so, we then proposed numerical investigations using Cimlib-CFD. To our knowledge, this was the first attempt to solve the single impingement jet cooling problem using the numerical methods developed in Cimlib-CFD. The overall observations on the results showed our ability to capture the specific turbulent flow behavior of the studied configuration. Indeed, the instantaneous velocity and pressure fields clearly showed the rebound of primary vortices on the plate creating secondary vortices in the opposite direction. With this analysis, we managed to explain the re-injection of cold fluid air on the plate and thus, the appearance of a secondary peak in the radial Nusselt number distribution.

Then, we proposed in Chapter 3 a set of numerical tools to solve aerothermal problems in Cimlib-CFD. These numerical tools made it possible for us to simulate highly convective heat transfer in turbulent flows. However, additional numerical models were needed to reinforce aerothermal coupling in the simulation. We thus proposed to implement two additional numerical models in our finite element library which are: (i) the Sutherland law and (ii) the Smagorinsky Model. A numerical benchmark study of a 3D ventilated cavity was then made in order to validate our set of numerical tools. The results were in good agreement with experimental and numerical references. In particular, they gave us the best way to couple the unsteady incompressible Navier-Stokes solver with the thermal convection-diffusion solver using a modified dynamic viscosity and a modified thermal conductivity.

In Chapter 4, we proposed an innovative adaptive multiscale method that allowed dealing with highly convective aerothermal computations in Cimlib-CFD. Indeed, this chapter aimed at proposing a new multiscale mesh adaptation technique that was well suited for the type of aerothermal problems developed in the first two chapters. To do so, we proposed a new a posteriori error estimate based on the VMS method for anisotropic adaptive fluid mechanic problems. This new error estimator was defined by the combination of both (i) an interpolation based anisotropic error indicator and (ii) a subscales error estimator. The subscales error estimator was computed using two different methods. The first one used stabilizing parameters derived from the VMS analysis and local norms defined on the elements. The second method used a linear combination of bubble functions to establish a pointwise computation of the error. The results showed that the method using bubble functions was better suited to drive both isotropic and anisotropic mesh adaptation. Then, the results on analytic 2D and 3D benchmarks also showed that the proposed multiscale mesh adaptation technique allowed to obtain highly precise solutions with much less elements in comparison with other mesh adaptation techniques. This multiscale mesh adaptation technique was then applied to the realistic industrial case of the single impingement jet cooling. On this complex industrial case, the new multiscale mesh adaptation technique showed promising qualitative results.

Finally, in Chapter 5, we proposed a description of the parallel software capabilities in Cimlib-CFD. To do so, we presented the mesh partitioning principles in the FE framework. Then, the algorithm behind parallel mesh adaptation was introduced as well as important notions on the particular treatment of the partitions' interfaces. Finally, we presented the two different hardware systems used during this PhD thesis. The first one was the lab's cluster, called Cluster Intel, allowing the development of numerical methods. The second one however, was the GENCI Occigen II supercomputer on which we executed Cimlib-CFD for the first time during this PhD work. This system allowed us to produce numerical results using massively parallel computations. The adaptation of Cimlib-CFD on this new system required, among others, the development of a checkpoint-restart procedure that we designed following a certain number of constrains imposed by the supercomputer user policy. Then, we proposed application cases for parallel computations. The first one was dedicated to a strong scalability analysis of our finite element code on a well known benchmark. The results showed a linear increase of the speed-up with the allocated resources allowing us to conclude on the hard strong scalability of the numerical methods used to solve linear systems in Cimlib-CFD. Parallel mesh adaptation was tested on the two hardware systems for the case of the single impingement jet cooling. In this work, we showed the parallel capabilities of the software by presenting dynamic parallel mesh adaptation over 28 cores on the Cluster Intel and then, over 504 cores on the GENCI Occigen II supercomputer. Our results showed that we obtained relevant partitions in both cases. Finally, we presented in this chapter a more complex but realistic industrial case. The latter concerned the cooling of a complete turbine vane composed by 39 holes. To study this geometry, an incremental space discretization was chosen along with adapted allocated resources. The computational times on these incremental meshes showed a linear behavior which demonstrated the good scalability of Cimlib-CFD on the supercomputer.

6.2 Perspectives

Along this work, several perspective were proposed. In this section, we propose to summarize the main improvement axis highlighted in this PhD work. There are mainly two axis: one concerning physical modeling, and the other on a more general numerical problematic.

6.2.1 Physical modeling of the single impinging jet cooling

Concerning the physical modeling of the single impingement jet cooling several perspectives were proposed for future works. In fact, the influence of the outlet boundary condition still seems to play a crucial role on the flow motion in the boundary layer. The location of this boundary condition and the way to impose it in the finite element framework is still an open question. Furthermore, the mesh in the boundary layer can still be improved in order to capture more precisely both (i) the velocity profiles in the boundary layer and (ii) the thermal gradient on the plate.

Furthermore, even if we are confident that the proposed synthetic perturbations model for the turbulent inlet boundary condition is, indeed, well suited for our simulation, other solutions can be explored. Among others, we propose for future works a preliminary computation of the flow inside a long inlet tube in order to provide the turbulent inlet boundary condition of a fully developed jet. It will then be interesting to analyze the effects of this inlet boundary condition on the Nusselt number distribution.

In addition, the works of Dairay et al. in [23] and Aillaud et al. in [19] propose to use a statistical analysis of the heat transfer events on the plate. Indeed, in order to deepened the analysis of the flow turbulence, it can be interesting to apply their method using Probability Density Function (PDF) to decompose the heat transfer and to analyze at which scale our results are correlated to theirs. Going even further, the use of conditional averaging procedure aiming at dissociating some heat transfer events in the computation of the mean radial Nusselt number distribution can also be a solution to look on.

Finally, from an industrial point of view, other types of cooling can be studied with Cimlib-CFD. In particular, important works on film cooling have already started and can be found in the literature. Studying the film cooling can, in fact, enable us to understand the other physical phenomenon occurring during the turbine blade cooling. In particular, it will allow us, in the future, to propose a complete modeling of the flow inside the turbine blade.

Related to this, on the complete turbine vane geometry and multi-jets simulation, a number of improvements can be drawn from this work. Indeed, there is still a need to improve this aerothermal simulation in order to capture the convective heat transfers inside the geometry. To do so, future works will have to answer questions related to the size of the mesh and control of errors in the aerothermal context. In particular, our numerical tools will have to demonstrate their capabilities on such complex 3D geometry and at a computation scale never achieved before. Achieving this performance will require performing aerothermal error estimates allowing to target directly the turbulent features of the flow motion but also, the distribution of the heat transfer inside the cavity. This will be done by associating error estimates from different PDE equations. We will developed this idea in the next section.

6.2.2 VMS error estimates for anisotropic adaptive simulations

Indeed, from a numerical point of view, the proposed multiscale mesh adaptation method showed promising results on analytic benchmarks in 2D and 3D. Thus, the next step of this work is to demonstrate the efficiency of this new mesh adaptation method on more realistic industrial cases. To do so, we will have to demonstrate *quantitatively* the capabilities of this new method. For instance, as soon as the physical modeling issues on the impingement jet cooling are dealt with, it will be interesting to compare the heat transfer distributions obtained on both (i) a multiscale adapted mesh and (ii) a large isotropic reference mesh. Doing so, as for analytic benchmarks, we will hopefully be able to show that, with our new multiscale mesh adaptation method, we manage to obtain equivalent aerothermal solutions using much less elements and thus, much less computational resources compared to standard mesh adaptation methods.

Secondly, the combination between the anisotropic local error indicator and the subscales error estimator could be done differently. Indeed, the proposition made in this thesis consisting of scaling the hessian's eigenvalues using a scale factor based on the subscale error was the first idea that came to our minds when addressing the problem. Furthermore, it showed very promising results and thus, it was interesting to follow this direction. However, one could combine these two information differently by doing another type of error analysis. For example, one could consider intersecting the two metric tensors $\mathscr{H}_{iso}$ and $\mathscr{H}_{aniso}$ and obtain a different mesh, comparable to the one we obtained.

Thirdly, as mentioned in Chapter 4 and in the previous section, the subscales error estimate can be applied to other types of equations. Important works on this subject can be found in the literature. In fact, the PDE dependency of the subscales error estimate implies that we need to develop a new error estimate when we change the problem's equation. As an example, it is possible to apply this error estimate to the incompressible Navier-Stokes equations. This time, the error estimator is computed on the vector variable $\mathbf{v}$ in order to have an estimation of $||\mathbf{v}'||$. The same was also tested on the variable p in order to have an estimation of $||p'||$. The computation of the error estimate using the residual free bubble matrix have yet to be undertaken for these equations, and the way to combine this error estimate with mesh adaptation will also be challenging.

Finally, this PhD work is really the first step into developing a complete aerothermal error estimate. Indeed, as mention in the previous section, the objective is then to control the errors simultaneously from the different PDEs solved during the turbine vane cooling (mainly the Navier-Stokes equations and the convection-diffusion equation). Indeed, a complete aerothermal error estimate will make it possible for us to control all the errors of the problem and to correlate them to one another in order to obtain the best mesh possible in these complex aerothermal simulations. This will require to decline subscales error estimates for all the equations, and to create an aerothermal module that will associate these error estimates. With these advanced possible combinations, we will hopefully be able to solve this kind of complex aerothermal problems at reduced computational costs.

Résumé du chapitre en français

Cette thèse de doctorat a été consacrée au développement d'une méthode élément finis adaptative permettant d'améliorer la simulation aérothermique du refroidissement des moteurs d'avion. Plus précisément, notre but a été de développer une méthode d'adaptation de maillage multi-échelle dédiée à la résolution des problèmes aérothermiques rencontrés, en particulier, dans le refroidissement des aubes de turbines. Pour étudier ce problème aérothermique complexe, il a été nécessaire de focaliser notre étude sur une configuration benchmark du refroidissement par jet impactant mettant en jeu des transferts thermiques hautement convectifs dans des écoulements turbulents.

Ce travail nous a permit de mieux comprendre les phénomènes physiques complexes intervenant lors du refroidissement par impact. A notre connaissance, il s'agit de la première tentative de résoudre ce problème aérothermique en utilisant les méthodes numériques développées au sein du CEMEF MINES-Paristech. L'analyse de nos résultats a montré notre capacité à capturer le comportement turbulent spécifique de cette configuration. En effet, l'analyse des champs macroscopiques de vitesse et pression a clairement montré le rebond des structures primaires sur la plaque, générant ainsi des tourbillons secondaires dans la direction opposée de l'écoulement. Avec cette analyse, nous avons pu expliquer la ré-injection d'air froid sur la plaque et donc, l'apparition d'un pique secondaire dans la distribution radiale du nombre de Nusselt.

Dans ce manuscrit, nous avons ensuite décrit plus particulièrement les outils numériques nous permettant de réaliser des simulations aérothermiques turbulentes avec Cimlib-CFD. De plus, nous avons proposé deux modèles additionnels permettant de renforcer le couplage aérothermique de la simulation. Ainsi, nous avons présenté l'implémentation (i) d'une loi de Sutherland et (ii) d'un modèle de Smagorinsky. Une étude de validation de ces deux modèles a été proposée sur un cas de cavité 3D ventilée. Les résultats sur ce cas de validation sont très proches des références expérimentales et numériques auquel nous nous comparons. En particulier, cette étude de validation nous a permit de choisir la meilleure manière de coupler le solver Navier-Stokes VMS incompressible et le solver thermique de convection-diffusion en utilisant une viscosité dynamique modifiée, ainsi qu'une conductivité thermique modifiée.

Dans le Chapitre 4, nous avons proposé une méthode d'adaptation de maillage innovante permettant de gérer des simulations aérothermiques hautement convectives avec Cimlib-CFD. En effet, le but de ce chapitre a été de proposer une nouvelle méthode d'adaptation de maillage, bien adaptée aux cas aérothermiques développés dans les deux premiers chapitres. Pour réaliser cela, nous avons proposé un nouvel estimateur d'erreur à posteriori basé sur la méthode VMS pour les problèmes de Mécanique des Fluides. Ce nouvel estimateur d'erreur a été défini par (i) un estimateur d'erreur d'interpolation anisotrope et (ii) un estimateur d'erreur sous-échelle.

Les résultats sur des cas analytiques 2D et 3D ont montré que la méthode d'adaptation de maillage multi-échelle permettait d'obtenir des solutions hautement précises avec un nombre réduit d'éléments en comparaison aux autres techniques d'adaptation de maillage. De plus, cette méthode d'adaptation de maillage multi-échelle a également été appliquée au cas plus réaliste du refroidissement par jet impactant et nous avons pu montré des résultats qualitatifs prometteurs sur cette configuration.

Enfin, nous proposons dans ce manuscrit une description des possibilités de calcul parallèle offertes par Cimlib-CFD. Nous présentons les principes de partitionnement de maillage par décomposition de domaine dans le contexte numérique des éléments finis. Ensuite, l'algorithme permettant une adaptation de maillage parallèle est introduit ainsi que le traitement particulier des interfaces entre les partitions. Les deux systèmes de calcul utilisés pendant ce doctorat sont également présentés. Le premier est un cluster interne au laboratoire, appelé Cluster Intel, qui a plus particulièrement été utilisé pour le développement des méthodes numériques présentées dans ce manuscrit. Le second en revanche, est le super-calculateur national GENCI Occigen II sur lequel nous avons pu exécuter Cimlib-CFD pour la première fois pendant cette thèse de doctorat. Ce système nous a permit de réaliser des calculs massivement parallèles mais a nécessité une adaptation de Cimlib-CFD à ce nouvel environnement de calcul. En particulier, une procédure d'arrêt et de reprise du calcul a été implémentée pour s'adapter aux contraintes imposées par la politique utilisateur de ce nouveau système. Des cas d'application des techniques de parallélisation sont ensuite proposés. Le premier cas est dédié à l'étude de la scalabilité forte des solvers linéaires utilisés pendant la thèse et reprend le fameux cas de la cavité ventilée 3D. Les résultats montrent une augmentation linéaire du speed-up avec l'augmentation des ressources de calcul. Ensuite, l'adaptation parallèle de maillage est testée sur les deux systèmes de calcul pour le cas du jet unique impactant. Dans ce travail, nous montrons les capacités parallèles du programme à deux échelles différentes.

Enfin, nous présentons dans ce chapitre un cas industriel plus réaliste qui concerne le refroidissement d'une aube de turbine complète composée de 39 trous. Pour étudier cette géométrie, nous proposons d'utiliser une discrétisation spatiale incrémentale en adaptant les ressources de calcul demandées à la taille du problème. Cela nous permet, dans une certaine mesure, de faire une analyse sur la scalabilité faible de notre code. Les temps de calcul sur ces maillages incrémentaux montrent une progression linéaire qui illustre le bon comportement de Cimlib-CFD sur le super-calculateur. De plus, les résultats montrent que nous sommes capable d'obtenir le champ de vitesse au sein de la géométrie. Cependant, des travaux futurs doivent être envisagés pour capture les coefficient de transferts thermiques. Pour cela, en comparant cette étude avec l'étude numériques menée sur le cas du jet unique et plus largement, par les études numériques réalisées dans d'autres laboratoire, il sera nécessaire d'augmenter la taille du problème et de lancer des calculs à plus grande échelle.

Nous proposons dans ce travail un certain nombre de perspectives concernant la modélisation physique du refroidissement par impact. En effet, l'influence de la condition aux limites de sortie semble toujours jouer un rôle crucial dans le déplacement du fluide, en particulier dans la couche limite. La localisation de cette condition aux limites et la manière de l'imposer dans Cimlib-CFD est toujours une question ouverte à ce jour. Ensuite, bien que nous soyons confiant sur le fait que la solution retenue pour imposer des perturbations synthétiques turbulentes à l'entrée du jet est la plus adaptée à notre contexte numérique, d'autres solutions peuvent être explorées. Parmi d'autres, il serait intéressant d'effectuer un calcul dans un long tube, de manière à fournir une condition aux limites turbulente et instationnaire en entrée de la simulation du refroidissement par jet impactant.

Un certain nombre de perspectives ont également été apportées aux travaux sur les estimateur d'erreur sous-échelle. En effet, la prochaine étape de ce travail sera de faire une étude quantitative sur les capacités de cette nouvelle méthode. Par exemple, dès lors que les problèmes liés à la modélisation physique du jet impactant seront réglés, il sera intéressant de pouvoir comparer les distribution de coefficient de transferts thermiques calculés sur (i) un maillage adaptés multi-échelle et (ii) un maillage isotrope de référence. Faisant cela, nous seront capables de montrer qu'il est possible d'obtenir des solutions aérothermiques équivalentes avec un nombre réduit d'éléments et donc, un nombre réduit de ressources de calcul.

Ensuite, comme mentionné dans le chapitre, l'estimateur d'erreur sous-échelle peut être appliqué à d'autres type d'équations. En effet, la dépendance de ce dernier à la PDE implique que nous devons développer un nouveau modèle pour chaque équation. Un travail sur l'application de cet estimateur d'erreur aux équations de Navier-Stokes incompressible a été engagé pendant ce doctorat. Cependant, le calcul de l'estimateur d'erreur utilisant des fonctions bulles doit encore être entrepris et la manière de la combiner à l'adaptation anisotrope de maillage peut s'avérer particulièrement difficile.

Finalement, ce travail de thèse est en réalité le premier pas vers le développement d'un estimateur aérothermique complet. En effet, l'objectif final serait de pouvoir contrôler les erreurs des différentes PDEs simultanément. Cela permettrait de corréler les erreurs du problème et de les contrôler en générant des maillages adaptés aux simulations aérothermiques de ce niveau de complexité.

Appendix A

List of Acronyms

2D two dimensional. 17, 25, 60, 64, 65, 67, 74, 80, 83, 84, 88, 89, 122, 124, 126

3D three dimensional. 10, 16, 17, 24, 25, 39, 42, 43, 50, 54, 55, 56, 58, 60, 61, 65, 71, 74, 80, 84, 88, 89, 98, 103, 105, 120, 122, 124, 126

AMR Adaptive Mesh Refinement. 9

BPR By-Pass Ratio. 2, 3

BSL BaSeLine. 19

CEMEF Centre de Mise En Forme des matériaux. 24, 98, 99, 125

CERFACS Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique. 17

CFD Computational Fluid Dynamic. 2, 7, 9, 10, 13, 16, 17, 39, 64, 88, 94, 98

CFL Computing and FLuid. 24

CFL Courant–Friedrichs–Lewy. 27

CG Conjugate Gradient. 93

CIFRE Convention Industrielle de Formation par la REcherche. 12

CINES Centre Informatique National de l'Enseignement Supérieur. 98, 100

CPU Central Processing Unit. 92, 94, 96, 98, 99, 111, 112

CSR Compress Sparse Row. 100

DARI Demande d'Attribution de Ressources Informatiques. 100

DNS Direct Numerical Simulation. 16

DOF Degrees Of Freedom. 94, 104

ERCOFTAC European Research Community on Flow, Turbulence and Combustion. 17, 28, 35

FDSM Filtered Dynamic Scale Model. 51

FE Finite Element. 11, 46, 74, 80, 92, 93, 99, 119, 123

FEM Finite Element Method. 46, 68, 93

FLOPS FLoating-point Operations Per Second. 9, 92, 98, 101

FPR Fan Pressure Ratio. 3, 12

GMRES Generalized Minimal RESidual. 93

HPC High Performance Computing. 2, 9, 16, 94, 97, 98, 100

LES Large Eddy Simulation. 7, 16, 51

MPI Message Passing Interface. 95

NASA National Aeronautics and Space Administration. 7, 9, 13, 92

NEO New Engine Option. 3

ONERA Office National d'Etudes et de Recherche Aérospatiales. 6, 16

PDE Partial-Differential Equation. 10, 27, 42, 46, 64, 88, 89, 93, 124, 125, 127

PDF Probability Density Function. 124

PETSc Portable Extensible Toolkit for Scientific computation. 93

PIV Particle Image Velocimetry. 116

RANS Reynolds-Averaged Navier-Stokes. 5, 7, 18, 19, 21, 24, 51

RST Reynolds Stress Transport. 7

SAE Safran Aircraft Engines. 2, 3, 4, 5, 6, 12, 106

SLURM Simple Linux Utility for Resource Management. 101, 103

SM Smagorinsky Model. 10, 48, 49, 51, 59

SPMD Single Program Multiple Data. 95

SpMV Sparse Matrix Vector product. 94, 99

SST Shear Stress Transport. 19, 21, 24

SUPG Streamline Upwind Petrov/Galerkin. 10, 32, 42, 46, 47, 61, 64, 66, 68, 69, 74, 78, 80, 108, 109

TET Turbine Entry Temperature. 4, 12

TRL Technology Readiness Level. 7, 9

URANS Unsteady Reynolds-Averaged Navier-Stokes. 5

VMS Variational Multi-Scale. 10, 32, 42, 43, 44, 47, 48, 53, 60, 61, 64, 65, 87, 88, 89, 95, 100, 103, 104, 108, 109, 114, 122, 125

VTU Visualization Toolkit Unstructured points data. 97

WALE Wall Adapting Local Eddy-viscosity. 59

ZDES Zonal Detached Eddy Simulation. 6

Appendix B

List of Symbols

Symbol	Description	Units	Page List
$(\cdot,\cdot)$	scalar product on the space $L^2(\Omega)$		43, 46, 47
$(\cdot,\cdot)_\Omega$	scalar product on the whole domain		44
$(\cdot,\cdot)_{\Omega_e}$	scalar product on the whole domain		44
$[\cdot]$	jump term		48
a	generic diffusion coefficient	$m^2.s^{-1}$	46, 68
A	constant in the modulation function	m^{-1}	27
$a(\cdot,\cdot)$	bilinear form		46, 47
a_1	constant of proportionality between the shear stress and the turbulent kinetic energy		20
a_{fluid}	fluid (air) thermal diffusivity	$m^2.s^{-1}$	42
A_m	amplitude of the excited azimuthal mode	$m.s^{-1}$	27
$a_\tau(\cdot,\cdot)$	stabilized bi-linear form		47
α	Level-Set function	m	49, 55
ave_i	average time of process i out of n_p processes	s	97
$\mathscr{B}$	operator acting on the boundary that defines the natural boundary condition		46
b_i	bubble functions		67
β	thermal expansion coefficient	K^{-1}	26, 50, 52
C	Sutherland's law constant	K	48
c_1	algorithmic constant in the continuity stabilizing parameter		45
c_2	algorithmic constant in the continuity stabilizing parameter		45
c_e	barycenter of the element		67
$\mathbf{c}_i$	centroid of the element		68
c_i^b	constant in the bubble functions combination		67
c_p	fluid (air) specific heat	$J.kg^{-1}.K^{-1}$	26, 50, 52
C'	constant from Almeida et al.		70
C	constant of proportionality independent of the element size		72
C_s	Smagorinsky constant		49
d	space dimension of the problem		42, 43, 67, 71
D	jet nozzle diameter (for a round jet)	m	16, 17, 18, 21, 24, 26, 27, 28, 30, 34, 35, 36, 39

Symbol	Description	Units	Page List
∇	gradient operator		42, 46
Δ	Laplacian operator		46
$\widehat{S_{ij}}$	symmetric part of the resolved velocity gradient on the mesh nodes	s^{-1}	49
Δ_e	length scale related to the grid size	m	48
e_h	exact discrete error		74, 75
e_i	eigenvectors of $\mathrm{H_R}$		70, 72
ϵ	turbulent dissipation	$\mathrm{J}.kg^{-1}.s^{-1}$	5, 19, 51
$\boldsymbol{\varepsilon}$	the strain rate tensor	$m.s^{-2}$	43
η_{Ω_e}	anisotropic local error indicator		71, 73
$\eta_{\Omega_e,new}$	new anisotropic local error indicator		73, 76, 80, 105
exp	exponential function		27
f	source term		66, 68, 74
f_0	impinging frequency of the large scale structures	$\mathrm{H}z$	27
$\mathrm{F_2}$	second blending function of the $k-\omega$ turbulence model		20
f_{mod}	modulation function		27
f_μ	Van Driest damping function		55
g	gravitational acceleration constant	$m.s^{-2}$	26, 50, 52
g	value of the Dirichlet boundary condition		46
g'	Green's function		66
$\mathrm{G}r$	Grashof number		50, 51
h	local mesh size		71
H	impact distance of the jet	m	16, 17, 18, 24, 26, 28, 39
$\mathcal{H}_{aniso}$	anisotropic metric tensor		70, 74, 83, 124
$\mathcal{H}^{new}_{aniso}$	new anisotropic metric tensor		10, 13, 65, 70, 74, 83, 84, 89
H_c	height of the conic shape in ANSYS CFX	m	18
h_{conv}	convective heat transfer coefficient	$\mathrm{W}.m^{-2}.\mathrm{K}^{-1}$	30, 108
h_d	unknown of the re-meshing problem		71
$h_{d,new}$	new unknown of the re-meshing problem		73
h_e	characteristic length of the element	m	45, 47
h_i	mesh sizes in the e_i directions		70
h_{in}	height of the inlet slot	m	51
$\mathcal{H}_{iso}$	new isotropic metric tensor		10, 13, 65, 69, 70, 72, 74, 83, 89, 124
h	Neumann boundary condition		46
h_{new}	size of the new mesh (after mesh adaptation)		72
h_{out}	height of the outlet slot	m	51
$\mathrm{H_R}$	recovered Hessian matrix		70, 71, 72
I	increment		32, 34
$\mathbf{I}_d$	d-dimensional identity tensor		43
I_{eff}	efficiency index		74, 75
k	turbulent kinetic energy	$\mathrm{J}.kg^{-1}$	5, 19, 21, 24, 51

Symbol	Description	Units	Page List
L_c	characteristic length of the problem	m	30, 50, 52
$\mathscr{L}$	generic differential operator		68
L^p	Lebesgue space of degree p		70, 71
L_{plate}	plate length	m	26
L_{tube}	tube length	m	26
α_{fluid}	fluid (air) thermal diffusivity	$m^2.s^{-1}$	26, 50, 52
λ	eigenvalue of the isotropic metric tensor		72
Λ	diagonal matrix of absolute value of the eigenvalues of $\mathrm{H}_\mathrm{R}(u_h(x))$		70
λ_d	diagonal matrix of absolute value of the eigenvalues of $\mathrm{H}_\mathrm{R}(u_h(x))$		71
λ_{fluid}	tabulated fluid (air) thermal conductivity	$\mathrm{W}.m^1.\mathrm{K}^1$	26, 30, 50, 52, 54, 60
λ_i	eigenvalues of $\mathrm{H}_\mathrm{R}(u_h(x))$		70
λ_{total}	calculated fluid thermal conductivity	$\mathrm{W}.m^1.\mathrm{K}^1$	30, 48, 50, 54, 56, 60, 109
λ_{turb}	turbulent fluid thermal conductivity	$\mathrm{W}.m^1.\mathrm{K}^1$	50
$\hat{\lambda}_i$	barycentric coordinates in the reference element		67
$\mathscr{M}$	metric field		70
$\dot{m}_{primary}$	mass flow rate of the primary flow	$kg.s^{-1}$	3
$\dot{m}_{secondary}$	mass flow rate of the secondary flow	$kg.s^{-1}$	3
max_i	maximum time of process i out of n_p processes	s	97
μ	generic dynamic viscosity	$\mathrm{P}a.s$	43
μ_{fluid}	fluid (air) dynamic viscosity	$\mathrm{P}a.s$	26, 43, 48, 52, 54, 59
μ_T	modified fluid dynamic viscosity	$\mathrm{P}a.s$	48, 54, 55, 56, 59
μ_{total}	total dynamic viscosity	$\mathrm{P}a.s$	48, 49, 54, 56, 60, 109, 114
μ_{turb}	turbulent fluid dynamic viscosity	$\mathrm{P}a.s$	48, 50, 53, 56
$\mathbf{n}$	normal direction of the walls		49, 55
$\mathscr{N}$	number of elements		43, 46, 76, 80, 84, 94
N_{interp}	number of interpolation points		68
$\mathscr{N}_{\Omega_h}$	number of elements for the triangulation Ω_h		70
$\mathscr{N}_{\Omega'_h}$	number of elements for the triangulation Ω'_h		71
$\mathrm{N}u$	Nusselt number		30
ν_{fluid}	fluid (air) kinematic viscosity	$m^2.s^{-1}$	26, 49, 50, 52
ν_t	eddy viscosity	$m^2.s^{-1}$	19
$\boldsymbol{\omega}$	flow vorticity	s^{-1}	34
Ω	spatial domain of the fluid		42, 43, 74
$\partial\Omega$	boundary of the spatial domain		42, 43
Ω_e	finite element		43, 44, 46, 48
$\|\Omega_e\|$	volume of the element Ω_e	m^3	48, 71
Ω_h	triangulation of the domain		70

Symbol	Description	Units	Page List
Ω_{ref}	reference element		67
ω	turbulent specific dissipation	s^{-1}	5, 19, 21, 24
$\Omega_{\mathbf{v}}$	skew-symmetric part of the velocity gradient tensor	s^{-1}	32
ω_y	flow vorticity around the Y axis	s^{-1}	34
p	pressure	Pa	19, 32, 42, 43, 48, 88, 108, 124
P0	variable defined on an mesh element (i.e. $\in \mathbb{P}_0$)		66, 68
P1	variable defined on a mesh node (i.e. $\in \mathbb{P}_1$)		68, 70
p_0	inlet & outlet static pressure	Pa	26
p_h	resolved part of the pressure	Pa	43
$\mathbb{P}_k$	space polynomial of degree k		46
p'	unresolved part of the pressure	Pa	43, 44, 88, 124
Pe_{Ω_e}	local Peclet number on element Ω_e		47
Φ_{conv}	heat flux imposed on the intrados and extrados	$W.m^{-2}$	108
ϕ_m	frequency of the excited azimuthal mode	$rad.s^{-1}$	27
Pr	Prandtl number		50, 53
q	weighting function for the pressure	Pa	44
Q	criterion for vortex identification		32, 34
q_h	resolved part of the weighting function for the pressure	Pa	44
q'	unresolved part of the weighting function for the pressure	Pa	44
r	distance to the jet center axis	m	18, 21, 26, 27, 30, 35, 36
R	jet nozzle radius	m	27
$\mathscr{R}$	orthogonal matrix built with the eigenvectors $(e_i)_{i=1,d}$ of $H_R(u_h(x))$		70, 72
R_c	radius of the conic shape in ANSYS CFX	m	18
$\mathscr{R}_C$	continuity residual		44
$\mathscr{R}_M$	momentum residual		44
Ra	Rayleigh number		50, 51
Re_{inj}	injection Reynolds number		16, 17, 26
ρ	generic density	$kg.m^{-3}$	42, 43
ρ_{fluid}	fluid (air) density	$kg.m^{-3}$	26, 42, 48, 52
S	invariant measure of the strain rate in the boundary layer		20
S_{Ω_e}	stretching factor		74, 75
$S_{\mathbf{v}}$	symmetric part of the velocity gradient tensor	s^{-1}	32, 49
σ	constant in the modulation function	m^{-1}	27
$\boldsymbol{\sigma}$	Cauchy stress tensor		42
St_D	Strouhal Number associated with the jet nozzle diameter D		27
t	current time	s	42
T_0	impinging period of the main large scale structures	s	27, 52

Symbol	Description	Units	Page List
Δt	time step of the simulation	s	27, 109
$\mathscr{T}_h$	partition of the domain		43, 44
T_{init}	initial temperature	K	25, 48
T_{inlet}	jet temperature	K	25, 26
T_{plate}	plate temperature	K	25, 26
T_{ref}	temperature far from wall	K	50, 51, 52, 108
t_{total}	total physical duration of the simulation	s	28, 42, 53, 54, 109
T_{wall}	temperature of the hot wall	K	50, 51, 52, 108
τ	shear stress in the boundary layer	Pa	19
τ_C	stabilizing parameter for the continuity residual		45
τ_e	stabilizing parameter in the SUPG scheme		47
τ_K	stabilizing parameter for the momentum residual		45
τ_w	wall shear stress	Pa	49, 55
θ	angle on the inlet plane	rad	27
θ_T	normalized temperature		20, 30
$time_i$	time spent by process i	s	97
u	solution of the convection-diffusion problem (temperature)	K	42, 47, 48, 65, 70, 74, 83
u_h	resolved part of the convection-diffusion solution (temperature)	K	47, 65, 68, 70, 71, 72, 74, 75, 80
u'	unresolved part of the convection-diffusion solution (temperature)	K	47, 65, 72
u'_{bub}	internal residual error		67, 68
u'_{poll}	inter-element error		67
$\|\|u'_{tol}\|\|_{L^\infty}$	tolerance value on the element in L^∞ norm		72
u'_{TOL}	tolerated pointwise error		72
u_τ	friction velocity at the nearest wall (usually the plate)	$m.s^{-1}$	49
$\mathbf{v}$	velocity vector	$m.s^{-1}$	19, 32, 42, 43, 46, 48, 68, 74, 88, 108, 124
V_b	bulk velocity	$m.s^{-1}$	26, 27, 34
V_c	centerline velocity	$m.s^{-1}$	26
$\mathbf{v}_h$	resolved part of the velocity vector	$m.s^{-1}$	43
V_{in}	velocity magnitude of the inlet flow	$m.s^{-1}$	51
$\mathbf{v}'$	unresolved part of the velocity vector	$m.s^{-1}$	43, 44, 88, 124
$\mathbf{w}$	weighting function for the velocity vector	$m.s^{-1}$	44
w	weighting function for the temperature	K	47
$\mathbf{w}_h$	resolved part of the weighting function for the velocity vector	$m.s^{-1}$	44
w_h	resolved part of the weighting function for the temperature	K	47

Symbol	Description	Units	Page List
$\mathbf{w}'$	unresolved part of the weighting function for the velocity vector	$m.s^{-1}$	44
w'	unresolved part of the weighting function for the temperature	K	47
$\mathbf{x}$	position vector	m	42, 66, 68
$\mathbf{x}_i$	interpolation points		68
$\mathbf{y}$	position vector	m	66
z	distance to the nearest wall (usually the plate)	m	30, 35, 49
z_1	height of the first layer on the plate	m	18, 20
z^+	dimensionless wall distance		18, 28, 49, 55, 56

List of Figures

List of Tables

Bibliography

[1] M. Hetmańczyk, L. Swadźba, and B. Mendala. “Advanced materials and protective coatings in aero-engines application”. *Journal of Achievements in Materials and Manufacturing Engineering* Vol. 24, nr 1 (2007), pp. 372–381. URL (*Cited on page* 2).

[2] O. Verseux and Y. Sommerer. “New Challenges for Engine Nacelle Compartments Pressure and Thermal Loads Management With Aircraft Engine Evolution”. St. Petersburg, Russia: Airbus Operation S.A.S, 2014 (*Cited on page* 3).

[3] A. Terzis, P. Ott, J. von Wolfersdorf, B. Weigand, and M. Cochet. “Detailed Heat Transfer Distributions of Narrow Impingement Channels for Cast-In Turbine Airfoils”. *Journal of Turbomachinery* 136.9 (2014), pp. 091011–091011–9. URL (*Cited on page* 4).

[4] L. Schwerdt, T. Hauptmann, A. Kunin, J. R. Seume, J. Wallaschek, P. Wriggers, L. P.-v. Scheidt, and S. Löhnert. “Aerodynamical and Structural Analysis of Operationally Used Turbine Blades”. *Procedia CIRP*. Proceedings of the 5th International Conference in Through-life Engineering Services Cranfield University, 1st and 2nd November 2016 59 (2017), pp. 77–82. URL (*Cited on page* 4).

[5] J.-C. Han, S. Dutta, and S. Ekkad. *Gas Turbine Heat Transfer and Cooling Technology, Second Edition*. CRC Press, 2012 (*Cited on page* 5).

[6] P. Aillaud. “Simulations aux grandes échelles pour le refroidissement d’aubages de turbine haute-pression”. PhD thesis. Toulouse: Institut National Polytechnique de Toulouse, 2017. URL (*Cited on pages* 5, 26).

[7] D. Thibault. “Etude du refroidissement par impact de jets à travers une paroi mince et avec un écoulement cisaillant amont: application aux aubes de turbines”. PhD thesis. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d’Aérotechique-Poitiers, 2009 (*Cited on pages* 5, 6).

[8] M. Huynh. *Bilan du Cas 1 Jet Unique sur plaque plane*. Intern Report Safran Aircraft Engines. Villaroche: SAFRAN Group, 2016 (*Cited on page* 5).

[9] E. Laroche and F. Chedevergne. *Rapport d’Avancement AETHER C*. Rapport Interne ONERA. Toulouse: ONERA, 2016 (*Cited on page* 6).

[10] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*. Tech. rep. 2014 (*Cited on pages* 7, 8, 64, 92, 93).

[11] S. Royal Institute of Technology KTH. *http://exaflow-project.eu Enabling Exascale Fluid Dynamics Simulation*. URL (*Cited on pages* 9, 93).

[12] J. Hoessler, J. Lombard, S Dhandapani, and S. Sherwin. “ExaFLOW use cases: Wing tip vortex, Imperial Front Wing and McLaren front section” (2016) (*Cited on page* 9).

[13] A. N. Brooks and T. J. R. Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. *Computer Methods in Applied Mechanics and Engineering* 32.1 (1982), pp. 199–259. URL (*Cited on pages* 10, 42).

[14] J. W. Baughn and S. Shimizu. “Heat Transfer Measurements From a Surface With Uniform Heat Flux and an Impinging Jet”. *Journal of Heat Transfer* 111.4 (1989), pp. 1096–1098. URL (*Cited on pages* 16, 24, 25, 28, 37).

[15] D. Cooper, D. C. Jackson, B. E. Launder, and G. X. Liao. “Impinging jet studies for turbulence model assessment—I. Flow-field experiments”. *International Journal of Heat and Mass Transfer* 36.10 (1993), pp. 2675–2684. URL (*Cited on pages* 16, 17, 20, 23, 24, 25, 26, 28, 35).

[16] C Cornaro, A. S Fleischer, and R. J Goldstein. “Flow visualization of a round jet impinging on cylindrical surfaces”. *Experimental Thermal and Fluid Science* 20.2 (1999), pp. 66–78. URL (*Cited on pages* 16, 28).

[17] M. Tsubokura, T. Kobayashi, N. Taniguchi, and W. P. Jones. “A numerical study on the eddy structures of impinging jets excited at the inlet”. *International Journal of Heat and Fluid Flow*. Selected Papers from the Fifth International Conference on Engineering Turbulence Modelling and Measurements 24.4 (2003), pp. 500–511. URL (*Cited on page* 16).

[18] M. Fenot, J. J. Vullierme, and E. Dorignac. “Local heat transfer due to several configurations of circular air jets impinging on a flat plate with and without semi-confinement”. *International Journal of Thermal Sciences* 44.7 (2005), pp. 665–675. URL (*Cited on pages* 16, 24, 28).

[19] P. Aillaud, F. Duchaine, and L. Gicquel. “LES of a Round Impinging Jet: Investigation of the Link Between Nusselt Secondary Peak and Near-Wall Vortical Structures”. American Society of Mechanical Engineers, 2016, V05BT11A002–V05BT11A002. URL (*Cited on pages* 16, 17, 24, 26, 27, 28, 30, 37, 123).

[20] P. Grenson and H. Deniau. “Large-Eddy simulation of an impinging heated jet for a small nozzle-to-plate distance and high Reynolds number”. *International Journal of Heat and Fluid Flow* 68 (2017), pp. 348–363. URL (*Cited on pages* 16, 26, 28, 30).

[21] N. Uddin, S. O. Neumann, and B. Weigand. “LES simulations of an impinging jet: On the origin of the second peak in the Nusselt number distribution”. *International Journal of Heat and Mass Transfer* 57.1 (2013), pp. 356–368. URL (*Cited on pages* 16, 59).

[22] T. Dairay, V. Fortuné, E. Lamballais, and L. E. Brizzi. “LES of a turbulent jet impinging on a heated wall using high-order numerical schemes”. *International Journal of Heat and Fluid Flow* 50 (2014), pp. 177–187. URL (*Cited on pages* 16, 27).

[23] T. Dairay, V. Fortuné, E. Lamballais, and L.-E. Brizzi. “Direct numerical simulation of a turbulent jet impinging on a heated wall”. *Journal of Fluid Mechanics* 764 (2015), pp. 362–394. URL (*Cited on pages* 16, 18, 24, 26, 27, 28, 30, 34, 123).

[24] P. Aillaud, F. Duchaine, L. Y. M. Gicquel, and S. Didorally. “Secondary peak in the Nusselt number distribution of impinging jet flows: A phenomenological analysis”. *Physics of Fluids* 28.9 (2016), p. 095110. URL (*Cited on pages* 18, 59).

[25] J. Wienand, A. Riedelsheimer, and B. Weigand. “Numerical study of a turbulent impinging jet for different jet-to-plate distances using two-equation turbulence models”. *European Journal of Mechanics - B/Fluids*. Rotating Flows 61, Part 2 (2017), pp. 210–217. URL (*Cited on pages* 18, 24, 27, 28).

[26] F. Menter. “Zonal Two Equation k-w Turbulence Models For Aerodynamic Flows”. In: *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*. American Institute of Aeronautics and Astronautics. URL (*Cited on page* 19).

[27] D. C. Wilcox. “Reassessment of the scale-determining equation for advanced turbulence models”. *AIAA Journal* 26.11 (1988), pp. 1299–1310. URL (*Cited on page* 19).

[28] B. E. Launder and B. I. Sharma. “Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc”. *Letters in Heat and Mass Transfer* 1.2 (1974), pp. 131–137. URL (*Cited on page* 19).

[29] Y. Yang, A. Bazile, Y. Mesri, M. Yang, and Y. Rao. “Aerothermal Impingement Jet Flow Simulation And Analysis For Aircraft Engines”. Bachelor’s Thesis. Shanghai Jiao Tong University, 2018 (*Cited on page* 20).

[30] M. Hadžiabdić and K. Hanjalić. “Vortical structures and heat transfer in a round impinging jet”. *Journal of Fluid Mechanics* 596 (2008), pp. 221–260. URL (*Cited on page* 26).

[31] R. Gautier, S. Laizet, and E. Lamballais. “A DNS study of jet control with microjets using an immersed boundary method”. *International Journal of Computational Fluid Dynamics* 28.6-10 (2014), pp. 393–410. URL (*Cited on page* 27).

[32] W. Bechara, C. Bailly, P. Lafon, and S. M. Candel. “Stochastic approach to noise modeling for free turbulent flows”. *AIAA Journal* 32.3 (1994), pp. 455–463. URL (*Cited on page* 27).

[33] C. Bogey, C. Bailly, and D. Juvé. “Noise Investigation of a High Subsonic, Moderate Reynolds Number Jet Using a Compressible Large Eddy Simulation”. *Theoretical and Computational Fluid Dynamics* 16.4 (2003), pp. 273–297. URL (*Cited on page* 27).

[34] E. Hachem, S. Feghali, R. Codina, and T. Coupez. “Anisotropic adaptive meshing and monolithic Variational Multiscale method for fluid–structure interaction”. *Computers & Structures*. Computational Fluid and Solid Mechanics 2013 122 (2013), pp. 88–100. URL (*Cited on page* 28).

[35] Y. Mesri, W. Zerguine, H. Digonnet, L. Silva, and T. Coupez. “Dynamic parallel adaption for three dimensional unstructured meshes: Application to interface tracking”. In: *Proceedings of the 17th International Meshing Roundtable*. Springer, 2008, pp. 195–212. URL (*Cited on pages* 28, 64, 70, 71, 73, 76).

[36] G. Haller. “An objective definition of a vortex”. *Journal of fluid mechanics* 525 (2005), pp. 1–26 (*Cited on page* 32).

[37] Y. Mesri, A. Bazile, J. C. Larroya-Huguet, and E. Hachem. “Parallel and adaptive VMS finite elements formulation for aerothermal problems”. *Computers & Fluids* 173 (2018), pp. 42–50. URL (*Cited on page* 36).

[38] T. J. R. Hughes, L. P. Franca, and M. Mallet. “A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics”. *Computer Methods in Applied Mechanics and Engineering* 54.2 (1986), pp. 223–234. URL (*Cited on page* 42).

[39] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. “A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations”. *Computer Methods in Applied Mechanics and Engineering* 73.2 (1989), pp. 173–189. URL (*Cited on page* 42).

[40] T. J. R. Hughes. “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods”. *Computer Methods in Applied Mechanics and Engineering* 127.1 (1995), pp. 387–401. URL (*Cited on pages* 42, 43, 47).

[41] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, and T. Coupez. “Stabilized finite element method for incompressible flows with high Reynolds number”. *Journal of Computational Physics* 229.23 (2010), pp. 8643–8665. URL (*Cited on pages* 42, 43, 44, 45).

[42] A. Ern and J.-L. Guermond. *Éléments finis: théorie, applications, mise en oeuvre*. Google-Books-ID: SHhnLz0EMywC. Springer Science & Business Media, 2002 (*Cited on pages* 43, 67).

[43] R. Codina. “Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods”. *Computer Methods in Applied Mechanics and Engineering* 190.13–14 (2000), pp. 1579–1599. URL (*Cited on page* 45).

[44] D. Irisarri and G. Hauke. “A posteriori pointwise error computation for 2-D transport equations based on the variational multiscale method”. *Computer Methods in Applied Mechanics and Engineering* 311 (2016), pp. 648–670. URL (*Cited on pages* 46, 64, 65, 67, 68, 69).

[45] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. “Advances in Stabilized Methods in Computational MechanicsThe variational multiscale method—a paradigm for computational mechanics”. *Computer Methods in Applied Mechanics and Engineering* 166.1 (1998), pp. 3–24. URL (*Cited on pages* 47, 66).

[46] E. Hachem, G. Jannoun, J. Veysset, and T. Coupez. “On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation”. *Applied Mathematics and Computation* 232 (2014), pp. 581–594. URL (*Cited on pages* 47, 64, 74).

[47] G. Hauke, M. H. Doweidar, and M. Miana. “The multiscale approach to error estimation and adaptivity”. *Computer Methods in Applied Mechanics and Engineering*. A Tribute to Thomas J.R. Hughes on the Occasion of his 60th Birthday 195.13–16 (2006), pp. 1573–1593. URL (*Cited on pages* 48, 66).

[48] W. Sutherland. “LII. The viscosity of gases and molecular force”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 36.223 (1893), pp. 507–531. URL (*Cited on page* 48).

[49] J. Smagorinsky. “General circulation experiments with the primitive equations”. *Monthly Weather Review* 91.3 (1963), pp. 99–164. URL (*Cited on page* 48).

[50] D. Lilly. “The representation of small scale turbulence in numerical simulation experiments”. 1967, pp. 195–210 (*Cited on page* 49).

[51] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. “A dynamic subgrid-scale eddy viscosity model”. *Physics of Fluids A: Fluid Dynamics* 3.7 (1991), pp. 1760–1765. URL (*Cited on page* 49).

[52] E. R. V. Driest. “On Turbulent Flow Near a Wall”. *Journal of the Aeronautical Sciences* 23.11 (1956), pp. 1007–1011. URL (*Cited on page* 49).

[53] P. V. Nielsen, A. Restivo, and J. H. Whitelaw. “The Velocity Characteristics of Ventilated Rooms”. *Journal of Fluids Engineering* 100.3 (1978), pp. 291–298. URL (*Cited on pages* 51, 56).

[54] Q. Chen. “Comparison of different k-epsilon models for indoor air flow computations”. *Numerical Heat Transfer, Part B: Fundamentals* 28.3 (1995), pp. 353–369. URL (*Cited on page* 51).

[55] W. Zhang and Q. Chen. “Large eddy simulation of indoor airflow with a filtered dynamic subgrid scale model”. *International Journal of Heat and Mass Transfer* 43.17 (2000), pp. 3219–3231. URL (*Cited on pages* 51, 53, 56).

[56] R. Ezzouhri, P. Joubert, F. Penot, and S. Mergui. “Large Eddy simulation of turbulent mixed convection in a 3D ventilated cavity: Comparison with existing data”. *International Journal of Thermal Sciences* 48.11 (2009), pp. 2017–2024. URL (*Cited on pages* 52, 53, 54).

[57] G. François. “Éléments finis stabilisés pour le remplissage en fonderie à haut Reynolds”. PhD thesis. École Nationale Supérieure des Mines de Paris, 2011. URL (*Cited on page* 53).

[58] S. Brogniez, C. Farhat, and E. Hachem. “A high-order discontinuous Galerkin method with Lagrange multipliers for advection–diffusion problems”. *Computer Methods in Applied Mechanics and Engineering* 264 (2013), pp. 49–66. URL (*Cited on page* 64).

[59] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. “High-order CFD methods: current status and perspective”. *International Journal for Numerical Methods in Fluids* 72.8 (2013), pp. 811–845. URL (*Cited on page* 64).

[60] R. Sevilla, A. J. Gil, and M. Weberstadt. "A high-order stabilised ALE finite element formulation for the Euler equations on deformable domains". *Computers & Structures*. UK Association of Computational Mechanics 181 (2017), pp. 89–102. URL (*Cited on page* 64).

[61] Q. Meng and M. Berzins. "Scalable large-scale fluid-structure interaction solvers in the Uintah framework via hybrid task-based parallelism algorithms: Uintah hybrid task-based parallelism algorithms". *Concurrency and Computation: Practice and Experience* 26.7 (2014), pp. 1388–1407. URL (*Cited on page* 64).

[62] D. Ibanez, I. Dunn, and M. S. Shephard. "Hybrid MPI-thread parallelization of adaptive mesh operations". *Parallel Computing* 52 (2016), pp. 133–143. URL (*Cited on page* 64).

[63] T. Panitanarak and S. M. Shontz. "A parallel log barrier-based mesh warping algorithm for distributed memory machines". *Engineering with Computers* (2017), pp. 1–18. URL (*Cited on page* 64).

[64] Y. Mesri, F. Alauzet, A. Loseille, L. Hascoët, B. Koobus, and A. Dervieux. "Continuous mesh adaptation models for CFD". *CFD Journal* 16.4 (2008), pp. 346–355. URL (*Cited on page* 64).

[65] G. Jannoun, E. Hachem, J. Veysset, J.-F. Zaragoci, and T. Coupez. "Fully space-time metric based anisotropic mesh adaptation for unsteady problems". *6th International Conference on Adaptive Modeling and Simulation, ADMOS 2013*. Lisbon, Portugal: International Center for Numerical Methods in Engineering, 2013, pp. 350–361. URL (*Cited on page* 64).

[66] W. Hassan and M. Picasso. "An anisotropic adaptive finite element algorithm for transonic viscous flows around a wing". *Computers & Fluids* 111 (2015), pp. 33–45. URL (*Cited on page* 64).

[67] E. Hachem, M. Khalloufi, J. Bruchon, R. Valette, and Y. Mesri. "Unified adaptive Variational MultiScale method for two phase compressible–incompressible flows". *Computer Methods in Applied Mechanics and Engineering* 308 (2016), pp. 238–255. URL (*Cited on page* 64).

[68] P. J. Frey and F. Alauzet. "Anisotropic mesh adaptation for CFD computations". *Computer Methods in Applied Mechanics and Engineering*. Unstructured Mesh Generation 194.48–49 (2005), pp. 5068–5082. URL (*Cited on page* 64).

[69] R. Aubry, S. Dey, E. Mestreau, and B. Karamete. "Boundary layer mesh generation on arbitrary geometries". Article in Press. 2017 (*Cited on page* 64).

[70] T. Coupez. "Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing". *Journal of Computational Physics* 230.7 (2011), pp. 2391–2405. URL (*Cited on page* 64).

[71] G. Jannoun, E. Hachem, J. Veysset, and T. Coupez. "Anisotropic meshing with time-stepping control for unsteady convection-dominated problems". *Applied Mathematical Modelling* 39.7 (2015), pp. 1899–1916. URL (*Cited on page* 64).

[72] R. C. Almeida, R. A. Feijóo, A. C. Galeao, C. Padra, and R. S. Silva. "Adaptive finite element computational fluid dynamics using an anisotropic error estimator". *Computer Methods in Applied Mechanics and Engineering* 182.3 (2000), pp. 379–400. URL (*Cited on pages* 64, 70).

[73] Y. Mesri, M. Khalloufi, and E. Hachem. "On optimal simplicial 3D meshes for minimizing the Hessian-based errors". *Applied Numerical Mathematics* (2016). URL (*Cited on page* 64).

[74] A. Loseille and F. Alauzet. "Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error". *SIAM Journal on Numerical Analysis* 49.1 (2011), pp. 38–60. URL (*Cited on page* 64).

[75] A. Agouzal and Y. V. Vassilevski. "Minimization of gradient errors of piecewise linear interpolation on simplicial meshes". *Computer Methods in Applied Mechanics and Engineering* 199.33–36 (2010), pp. 2195–2203. URL (*Cited on page* 64).

[76] P. Labbé, J. Dompierre, M.-G. Vallet, and F. Guibault. “Verification of three-dimensional anisotropic adaptive processes”. *International Journal for Numerical Methods in Engineering* 88.4 (2011), pp. 350–369. URL (*Cited on page* 64).

[77] Y. Mesri, H. Guillard, and T. Coupez. “Automatic coarsening of three dimensional anisotropic unstructured meshes for multigrid applications”. *Applied Mathematics and Computation* 218.21 (2012), pp. 10500–10519. URL (*Cited on page* 64).

[78] L. Kamenski and W. Huang. “How a Nonconvergent Recovered Hessian Works in Mesh Adaptation”. *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1692–1708. URL (*Cited on page* 64).

[79] B. N. Granzow, M. S. Shephard, and A. A. Oberai. “Output-based error estimation and mesh adaptation for variational multiscale methods”. *Computer Methods in Applied Mechanics and Engineering* 322 (2017), pp. 441–459. URL (*Cited on page* 64).

[80] J. Baiges and R. Codina. “Variational Multiscale error estimators for solid mechanics adaptive simulations: An Orthogonal Subgrid Scale approach”. *Computer Methods in Applied Mechanics and Engineering* (). URL (*Cited on page* 64).

[81] V. John and J. Novo. “A robust SUPG norm a posteriori error estimator for stationary convection–diffusion equations”. *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 289–305. URL (*Cited on page* 64).

[82] *Multiscale multi-dimensional explicit a-posteriori error estimation for fluid dynamics | TU Delft Repositories*. URL (*Cited on page* 64).

[83] G. Hauke, D. Fuster, and M. H. Doweidar. “Variational multiscale a-posteriori error estimation for multi-dimensional transport problems”. *Computer Methods in Applied Mechanics and Engineering* 197.33–40 (2008), pp. 2701–2718. URL (*Cited on page* 64).

[84] D. Irisarri and G. Hauke. “Pointwise Error Estimation for the One-Dimensional Transport Equation Based on the Variational Multiscale Method”. *International Journal of Computational Methods* (2016), p. 1750040. URL (*Cited on page* 64).

[85] A. Papastavrou and R. Verfürth. “A posteriori error estimators for stationary convection–diffusion problems: a computational comparison”. *Computer Methods in Applied Mechanics and Engineering* 189.2 (2000), pp. 449–462. URL (*Cited on page* 65).

[86] L. Formaggia, S. Perotto, and P. Zunino. “An anisotropic a-posteriori error estimate for a convection-diffusion problem”. *Computing and Visualization in Science* 4.2 (2001), pp. 99–104. URL (*Cited on page* 65).

[87] M. Picasso. “Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives”. *Computer Methods in Applied Mechanics and Engineering* 196.1–3 (2006), pp. 14–23. URL (*Cited on page* 65).

[88] I. Babuška and T. Strouboulis. *The Finite Element Method and Its Reliability*. Google-Books-ID: _Gv9KMOrA7sC. Clarendon Press, 2001 (*Cited on page* 71).

[89] G. Hauke, M. H. Doweidar, and S. Fuentes. “Mesh adaptivity for the transport equation led by variational multiscale error estimators”. *International Journal for Numerical Methods in Fluids* 69.12 (2012), pp. 1835–1850. URL (*Cited on page* 72).

[90] Z. Zhang. “Finite element superconvergence on Shishkin mesh for 2-D convection-diffusion problems”. *Mathematics of Computation* 72.243 (2003), pp. 1147–1177. URL (*Cited on page* 74).

[91] G. Houzeaux, R. Borrell, Y. Fournier, M. Garcia-Gasulla, J. H. Göbbert, E. Hachem, V. Mehta, Y. Mesri, and H. O. a. M. Vázquez. “High-Performance Computing: Dos and Don’ts”. *Computational Fluid Dynamics - Basic Instruments and Applications in Science*. InTech (2018), pp. 3–41. URL (*Cited on pages* 92, 93, 97).

[92] Y. Saad and M. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869. URL (*Cited on page* 93).

[93] J.-C. Jouhaud. *Simulations Aérodynamiques et Aérothermiques Avancées pour la Conception et l'Amélioration des Avions*. Toulouse: CERFACS, 2008 (*Cited on page* 93).

[94] I. Babuvška and W. Rheinboldt. "A-posteriori error estimates for the finite element method". *International Journal for Numerical Methods in Engineering* 12.10 (1978), pp. 1597–1615. URL (*Cited on page* 93).

[95] I. Babuvška and W. Rheinboldt. "Error Estimates for Adaptive Finite Element Computations". *SIAM Journal on Numerical Analysis* 15.4 (1978), pp. 736–754. URL (*Cited on page* 93).

[96] O. C. Zienkiewicz and J. Z. Zhu. "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique". *International Journal for Numerical Methods in Engineering* 33.7 (1992), pp. 1331–1364. URL (*Cited on page* 93).

[97] O. C. Zienkiewicz and J. Z. Zhu. "The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity". *International Journal for Numerical Methods in Engineering* 33.7 (1992), pp. 1365–1382. URL (*Cited on page* 93).

[98] M. Picasso. "A stopping criterion for the conjugate gradient algorithm in the framework of anisotropic adaptive finite elements". *Communications in Numerical Methods in Engineering* 25.4 (2009), pp. 339–355. URL (*Cited on page* 93).

[99] G. Manzinali, E. Hachem, and Y. Mesri. "Adaptive stopping criterion for iterative linear solvers combined with anisotropic mesh adaptation, application to convection-dominated problems". *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 864–880. URL (*Cited on page* 93).

[100] C. Walshaw, M. Cross, and M. G. Everett. "Parallel Dynamic Graph Partitioning for Adaptive Unstructured Meshes". *Journal of Parallel and Distributed Computing* 47.2 (1997), pp. 102–108. URL (*Cited on page* 93).

[101] H. L. De Cougny and M. S. Shephard. "Parallel refinement and coarsening of tetrahedral meshes". *International Journal for Numerical Methods in Engineering* 46.7 (1999), pp. 1101–1125. URL (*Cited on pages* 93, 95).

[102] T. Coupez, H. Digonnet, and R. Ducloux. "Parallel meshing and remeshing". *Applied Mathematical Modelling*. Dynamic load balancing of mesh-based applications on parallel 25.2 (2000), pp. 153–175. URL (*Cited on pages* 93, 95).

[103] Y. Mesri, H. Digonnet, and H. Guillard. "Mesh partitioning for parallel computational fluid dynamics applications on a grid". *Finite Vol. for Complex App* (2005). URL (*Cited on pages* 93, 96, 97, 98).

[104] Y. Mesri, H. Digonnet, and T. Coupez. "Advanced parallel computing in material forming with CIMLib". *Revue Européenne de Mécanique Numérique/European Journal of Computational Mechanics* 18.7-8 (2009), Pages 669–694. URL (*Cited on page* 93).

[105] Y. Mesri, H. Digonnet, and T. Coupez. "Hierarchical adaptive multi-mesh partitioning algorithm on heterogeneous systems" (2010), pp. 299–306. URL (*Cited on pages* 93, 96).

[106] R. Said, N. P. Weatherill, K. Morgan, and N. A. Verhoeven. "Distributed parallel Delaunay mesh generation". *Computer Methods in Applied Mechanics and Engineering* 177.1 (1999), pp. 109–125. URL (*Cited on page* 95).

[107] G. Houzeaux and R. Codina. "Transmission conditions with constraints in finite element domain decomposition methods for flow problems". *Communications in Numerical Methods in Engineering* 17.3 (2001), pp. 179–190. URL (*Cited on page* 96).

[108] F. Alauzet and M. Mehrenberger. “P1-conservative solution interpolation on unstructured triangular meshes”. *International Journal for Numerical Methods in Engineering* 84.13 (2010), pp. 1552–1588. URL (*Cited on page* 96).

[109] P. Farrell, M. Piggott, C. Pain, G. Gorman, and C. Wilson. “Conservative interpolation between unstructured meshes via supermesh construction”. *Computer Methods in Applied Mechanics and Engineering* 198.33-36 (2009), pp. 2632–2642 (*Cited on page* 96).

[110] P. E. Farrell and J. R. Maddison. “Conservative interpolation between volume meshes by local Galerkin projection”. *Computer Methods in Applied Mechanics and Engineering* 200.1 (2011), pp. 89–100. URL (*Cited on page* 96).

[111] F. Alauzet. “A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes”. *Computer Methods in Applied Mechanics and Engineering* 299 (2016), pp. 116–142 (*Cited on page* 96).

[112] C. Bahbah, Y. Mesri, and E. Hachem. “Interpolation with restrictions in an anisotropic adaptive finite element framework”. *Finite Elements in Analysis and Design* 142 (2018), pp. 30–41. URL (*Cited on page* 96).

[113] N. Capit, G. D. Costa, Y. Georgiou, G. Huard, C. Martin, G. Mounie, P. Neyron, and O. Richard. “A batch scheduler with high level components”. *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.* Vol. 2. 2005, 776–783 Vol. 2 (*Cited on page* 99).

[114] A. B. Yoo, M. A. Jette, and M. Grondona. “SLURM: Simple Linux Utility for Resource Management”. *Job Scheduling Strategies for Parallel Processing.* Ed. by D. Feitelson, L. Rudolph, and U. Schwiegelshohn. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 44–60 (*Cited on page* 101).

[115] M. Fénot and E. Dorignac. “Heat transfer and flow structure of an impinging jet with upstream flow”. *International Journal of Thermal Sciences* 109.Supplement C (2016), pp. 386–400. URL (*Cited on pages* 106, 108).

[116] E. Laroche, M. Fenot, E. Dorignac, J.-J. Vuillerme, L. E. Brizzi, and J. C. Larroya. “A Combined Experimental and Numerical Investigation of the Flow and Heat Transfer Inside a Turbine Vane Cooled by Jet Impingement” (2017), V05AT11A013. URL (*Cited on pages* 106, 107, 108, 109, 112, 116, 117, 118).

RÉSUMÉ

Considérant les récents progrès dans le domaine du Calcul Haute Perfomance, le but ultime des constructeurs aéronautiques tels que Safran Aircraft Engines (SAE) sera de simuler un moteur d'avion complet, à l'échelle 1, utilisant la mécanique des fluides numérique d'ici 2030. Le but de cette thèse de doctorat est donc de donner une contribution scientifique à ce projet. En effet, ce travail est consacré au développement d'une méthode élément finis variationnelle adaptative visant à améliorer la simulation aérothermique du refroidissement des aubes de turbine. Plus précisément, notre objectif est de développer une nouvelle méthode d'adaptation de maillage multi-échelle adaptée à la résolution des transferts thermiques hautement convectifs dans les écoulements turbulents.

Pour cela, nous proposons un contrôle hiérarchique des erreurs, basé sur des estimateurs d'erreur sous-échelle de type VMS. La première contribution de ce travail est de proposer une nouvelle méthode d'adaptation de maillage isotrope basée sur ces estimateurs d'erreur sous-échelle. La seconde contribution est de combiner (i) un indicateur d'erreur d'interpolation anisotrope avec (ii) un estimateur d'erreur sous-échelle pour l'adaptation anisotrope de maillage. Les résultats sur des cas analytiques 2D et 3D montrent que la méthode d'adaptation de maillage multi-échelle proposée nous permet d'obtenir des solutions hautement précises utilisant moins d'éléments, en comparaison avec les méthodes d'adaptation de maillage traditionnelles.

Enfin, nous proposons dans cette thèse une description des méthodes de calcul parallèle dans Cimlib-CFD. Ensuite, nous présentons les deux systèmes de calcul utilisés pendant le doctorat. L'un d'eux est, en particulier, le super-calculateur GENCI Occigen II qui nous a permit de produire des résultats numériques sur un cas d'aube de turbine complète composé de 39 trous en utilisant des calculs massivement parallèles.

MOTS CLÉS

Mécanique des Fluides, Simulations multi-échelles, Estimation d'erreur, Adaptation de maillage anisotrope, Calculs massivement parallèles, Refroidissement par jets impactant.

ABSTRACT

By 2030, considering the progress of HPC, aerospace manufacturers like SAE, hope to be able to simulate a whole aircraft engine, at full scale, using CFD. The goal of this PhD thesis is to bring a scientific contribution to this research framework. Indeed, the present work is devoted to the development of a variational adaptive finite element method allowing to improve the aerothermal simulations related to the turbine blade cooling. More precisely, our goal is to develop a new multiscale mesh adaptation technique, well suited to the resolution of highly convective heat transfers in turbulent flows.

To do so, we propose a hierarchical control of errors based on recently developed subscales VMS error estimators. The first contribution of this work is then to propose a new isotropic mesh adaptation technique based on the previous error estimates. The second contribution is to combine both (i) the coarse scales interpolation error indicator and (ii) the subscales error estimator for anisotropic mesh adaptation. The results on analytic 2D and 3D benchmarks show that the proposed multiscale mesh adaptation technique allows obtaining highly precise solutions with much less elements in comparison with other mesh adaptation techniques.

Finally, we propose in this thesis a description of the parallel software capabilities of Cimlib-CFD. Then, we present the two hardware systems used during this PhD thesis. The first one is the lab's cluster allowing the development of numerical methods. The second one however, is the GENCI Occigen II supercomputer which allows producing numerical results using massively parallel computations. In particular, we present a more realistic industrial concerning the cooling of a complete turbine vane composed by 39 holes.

KEYWORDS

Computational Fluid Dynamic, Variational Multi-Scale, Error Estimator, Anisotropic Mesh Adaptation, High Performance Computing, Impingement Jet Cooling.