

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLEB BLIDA



Institut d'aéronautique et des études spatiales

Mémoire de fin d'études pour l'obtention du diplôme

MASTER en aéronautique

Option : Propulsion avions

Thème

=====
Simulations numériques des écoulements internes et externes
avec
OpenFOAM
=====

Réalisée Par :

BAMBRIK Mohammed El Habib

Encadreur :

Mr ABADA Omar

Co- Encadreur :

Mr BENTRAD Hocine

Blida, Septembre 2015

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Avant tout je tiens à remercier ALLAH tout puissant de m'avoir donné cette foi et cette confiance ainsi que cette volonté et ce courage pour atteindre cette étape importante de ma vie.

J'adresse également mes remerciements à mes encadreurs Mr. ABADA Omar et Mr. BENTRAD Houcine pour leur soutien, leur conseils pertinents et leur aides précieuse, ainsi que la confiance qu'ils m'ont toujours témoignée tout au long de ce projet.

J'exprime mes remerciements aux membres du jury de m'avoir honoré en acceptant de juger ce travail.

je tiens aussi à témoigner mes sincère gratitude à tous ceux qui m'ont assistés de près ou de loin, par leurs encouragements, leurs conseils et leurs critiques.

Dédicaces

Je dédie ce travail :

Spécialement à mes parents à qui je dois énormément, qui ont cru en moi et qui m'ont

donné les moyens d'aller aussi loin.

A celle qui ma mi au monde et accompagnée durant les moments difficiles de long

cursus scolaire et universitaire.

A mes frères ILYES et AMINE

À tous ceux qui me sont chers et à leurs familles.

A mes amis AMIR, HOUSSAM, NACEREDDINE, SEIF, KHIREDDINE NABIH,

BEDREDDINE, SID AHMED, AMIR et RAMI.

Merci pour tous

MOHAMMED

Résumé

OpenFOAM est un outil open-source orienté CFD, intelligent et puissant fonctionnant sous Linux. Cet outil gratuit est capable de résoudre des écoulements instationnaires, compressibles ou multi-phases, mettant en jeu de la combustion ou de la cavitation, aussi rapidement que des logiciels commerciaux. C'est pourquoi cet outil est très intéressant pour les entreprises qui veulent réduire leurs coûts de licence de logiciels CFD.

En contrepartie, OpenFOAM est un outil complexe à mettre en œuvre, ce qui peut décourager certains utilisateurs potentiels à se tourner vers lui.

Ce document propose une bref introduction à OpenFOAM avec quelque cas de simulations comparés aux résultats théoriques et pratiques afin de démontrer sa fiabilité, et de donner au lecteur les informations nécessaires qui lui permettront d'utiliser OpenFOAM dans les meilleures conditions, même s'il ne connaît pas Linux.

Mots-clefs: OpenFOAM, La CFD open source, snappyHexMesh, blockMesh, Paraview, bash, FORTRAN, Linux.

Abstract

OpenFOAM is an open-source tool CFD oriented, smart and powerful running on Linux. This free tool is able to solve unsteady flows, compressible or multiphase involving combustion or cavitations, as quickly as commercial software. That's why this tool is very interesting for companies who wants to reduce their costs of CFD software licenses.

In return, OpenFOAM is a complex tool to implement, which may discourage some potential users to turn to it.

This document provides a brief introduction to OpenFOAM with some simulation cases compared to the theoretical and practical results to demonstrate its reliability, and give the reader the crucial information that will allow him to use OpenFOAM in the best conditions, even if he has no knowledge about Linux.

Keywords: OpenFOAM, the open source CFD, snappyHexMesh, blockMesh, Paraview, bash, FORTRAN, Linux.

ملخص

OpenFOAM هو أداة مفتوحة المصدر ذكية وقوية موجهة من اجل المحاكاة "CFD" يشتغل على لينكس. هذه الأداة المجانية قادرة على حل مجموعة من التدفقات المتقلبة للموائع القابلة للضغط أو متعددة الأطوار مع الاحتراق أو التكيف، بنفس سرعة البرمجيات التجارية. السبب الذي جعل هذه الأداة مثيرة للاهتمام عند الشركات التي ترغب في تقليل تكاليف ترخيص برامج CFD الخاصة بهم.

في المقابل OpenFOAM هو أداة معقدة، مما قد يحبط بعض المستخدمين لاستعماله.

تقدم هذه الأطروحة مقدمة OpenFOAM مع بعض المحاكاة المقارنة مع النتائج النظرية والعملية لإثبات موثوقيتها، وإعطاء القارئ المعلومات الأساسية التي من شأنها أن تسمح له باستخدام OpenFOAM في أفضل الظروف، حتى لو كان المستخدم ليس على دراية بلينكس.

الكلمات المفتاحية: OpenFOAM، المصدر المفتوح، المحاكاة بالحاسوب، snappyHexMesh، blockMesh، Paraview، باش، FORTRAN، لينكس.

NOMENCLATURE

Abréviations

OpenFOAM: Open Field Operation and Manipulation.

CFD	:	Computational fluid dynamics.
SGI	:	Silicon Graphics, Inc.
GPL	:	GNU Public Licence.
GNU	:	GNU is Not Unix.
sh	:	Bourne shell.
ksh	:	Korn shell.
bash	:	Bourne Again Shell.
STL	:	Stereo Lithography.
OBJ	:	3D OBJect Geometry.
Fortran	:	FORmula TRANslator.
VTK	:	Visualization Toolkit.
ASCII	:	American Standard Code for Information Interchange.
Oracle RDBMS	:	Object-Relational Database Management System.

Symboles grecs

α	:	Angle d'attaque [$^{\circ}$].
Δt	:	Accroissement en temps [s].
ρ	:	Masse volumique [kg/m^3].
ρ_0	:	Masse volumique du fluide dans le réservoir [kg/m^3].
μ	:	Viscosité dynamique [$\text{kg}/(\text{m}\cdot\text{s})$].
μ_t	:	Viscosité turbulente [$\text{kg}/(\text{m}\cdot\text{s})$].
ν	:	Viscosité cinématique [m^2/s].
ε	:	Dissipation de l'énergie cinétique turbulente [m^2/s^3].
ω	:	Dissipation spécifique de l'énergie cinétique turbulente [s^{-1}].

Symboles latins

A^*	:	Section au col de la tuyère [m^2].
A_e	:	Section d'entrer de la tuyère [m^2].
A_s	:	Section de sortie de la tuyère [m^2].
c	:	La corde moyenne du profil [m].
C_x	:	Coefficient de traîné.
C_z	:	Coefficient de portance.
C_p	:	Coefficient de pression.
D_T	:	Diffusivité thermique [m^2/s].
D	:	Le diamètre du cylindre [m].
f	:	La fréquence des oscillations [s^{-1}].
F_x	:	Force de trainée [N].
F_z	:	Force de portance [N].
Ma	:	Nombre de Mach [N].
M^*	:	Nombre de Mach au col.
M_s	:	Nombre de Mach a la sortie.
p	:	Pression cinématique [m^2/s^2].
P^*	:	Pression au col [Pa].
P_a	:	Pression statique a l'infini [Pa].
P_c	:	Pression génératrice [Pa].
P_0	:	Pression du fluide dans le réservoir [Pa].
k	:	Énergie cinétique du mouvement turbulent [m^2/s^2].
r	:	Constante des gaz parfaits 287,058 [$m^2/(kg.s^2)$].
Re	:	Le nombre de Reynolds.
St	:	Le nombre de Strouhal.
t	:	Le temps [s].
T	:	Température absolue [K].

T_0	:	Température du fluide dans le réservoir [K].
T^*	:	Température au col [K].
U	:	Vecteur vitesse [m/s].
V_∞	:	Valeur de la vitesse à l'infini [m/s].
x, y, z	:	Coordonnées cartésiennes [m].

TABLE DES MATIERES

Remerciements

Dédicaces

Résumé

Abstract

Nomenclature

Table des matières

Introduction générale..... 1

CHAPITRE 1

PRESENTATION D'OPENFOAM

1.1. Introduction.....	3
1.1.1. Histoire.....	4
1.1.2. Domaines d'applications.....	4
1.2. Quelques définitions.....	4
1.3. Les fonctionnalités d'OpenFOAM.....	5
1.4. Les avantages et les inconvénients d'OpenFOAM.....	6
1.4.1. Les Avantages.....	7
1.4.2. Les inconvénients d'OpenFOAM.....	8
1.5. La logique d'OpenFOAM.....	9
1.5.1. Les dossiers temporels « 0 ».....	10
1.5.2. Le dossier « constant »	10
1.5.3. Le dossier « system ».....	11
1.5.4 Les fichiers d'OpenFOAM.....	12
1.6. Les outils d'OpenFOAM.....	14
1.6.1. Pré-procession.....	15
1.6.2. Les solveurs.....	21
1.6.3. Post-traitement « Paraview ».....	23
1.7. Le dossier « tutorials ».....	25
1.7.1 Présentation.....	26
1.7.2. Lancer OpenFOAM dans le Terminal.....	28
1.7.3. Convertir un maillage.....	28
1.7.4. Lancer le calcul.....	29
1.7.5. Visualisation des résultats sous Paraview.....	30
1.7.6. Exportation des résultats.....	31
1.7.7. Auto exécution, les scripts Allrun et Allclean	32
1.8. Conclusion	32

CHAPITRE 2

ÉCOULEMENT LAMINAIRE BIDIMENSIONNEL AUTOUR D'UN CYLINDRE

2.1. Introduction.....	33
2.2. Physique du problème.....	34
2.3. Equations gouvernant l'écoulement.....	38
2.4. Résolution numérique.....	39
2.4.1. Prétraitement.....	39
2.4.2. Préparation du fichier blockMeshDict.....	41
2.4.3. Le traitement.....	44
2.4.3.1. Choix du solveur pisoFoam.....	44
2.4.3.2. Les conditions aux limites.....	45
2.4.4. Post-traitement.....	46
2.4.5. Lignes de courants à $Re = 2, 20$ et 100	47
2.5. Conclusion.....	50

CHAPITRE 3

ÉCOULEMENT AUTOUR D'UN PROFIL D'AILE

3.1. Introduction.....	51
3.2. Description physique.....	52
3.3. Le profil de Wortmann FXS02-196.....	53
3.3.1. Partie expérimentale.....	54
3.3.1.1. Description de la soufflerie.....	55
3.3.1.2. Description de l'expérience.....	58
3.3.1.3. Résultats obtenus.....	58
3.3.2. Partie numérique.....	60
3.3.2.1. Présentation.....	60
3.3.2.2. Prétraitement.....	61
3.3.2.3. Traitement.....	64
3.3.2.4. Post-traitement.....	66
3.3.2.5. Comparaison des Résultats.....	68
3.4. Le profil NACA4412.....	69
3.4.1. Présentation.....	70
3.4.2. Prétraitement.....	71
3.4.3. Traitement.....	72
3.4.4. Post-traitement.....	74
3.5. Conclusion.....	82

CHAPITRE 4

ÉCOULEMENT AUTOUR D'UN AVION SKYWALKER AVEC UN MAILLAGE DYNAMIQUE

4.1. Introduction.....	83
4.2. Description physique du problème.....	84
4.3. Définition d'un maillage dynamique.....	85
4.4. Présentation de l'utilitaire de maillage « snappyHexMesh »	85
4.5. Le drone Skywalker.....	85

4.6. Résolution numérique.....	86
4.6.1. Prétraitement.....	89
4.6.2. Traitement.....	94
4.6.2.1. Choix du solveur pimpleDyMFoam.....	94
4.6.2.2. Conditions aux limites.....	99
4.6.3. Post-traitement.....	100
4.7. Conclusion.....	109

CHAPITRE 5

ÉCOULEMENT SUPERSONIQUE DANS UN DIVERGEANT

5.1. Introduction.....	110
5.2. Description physique.....	111
5.2.1. Écoulement subsonique.....	112
5.2.2. Écoulement sonique adapté.....	112
5.2.3. Écoulement sonique non adapté.....	112
5.2.4. Écoulement supersonique adapté.....	113
5.2.5. Écoulement supersonique non adapté.....	113
5.3. Présentation du problème.....	114
5.4. Prétraitement.....	115
5.4.1. Génération des maillages.....	116
5.5. Traitement.....	120
5.6. Post-traitement.....	120
5.6.1. Différents résultats.....	122
5.6.1.1. Comparaison des paramètres thermodynamiques entre différentes géométries.....	122
5.7. Conclusion.....	130

Conclusion générale.....	131
---------------------------------	------------

Perspectives

Références bibliographique

Liste des figures

Annexes

INTRODUCTION GENERALE

OpenFOAM est un outil open source orienté CFD intéressant à plusieurs égards : c'est une bibliothèque qui peut être rapidement mise en place, large et évolutive, qui ne peut que s'étoffer et se perfectionner au cours du temps à l'aide de la communauté d'utilisateurs et de programmeurs indépendants qui entourent ce projet ; c'est également un outil inédit qui permet, en plus de la CFD, de faire des maillages ou d'importer des maillages faits par d'autres logiciels. Quant au post-traitement des résultats de calcul d'OpenFOAM, il peut se faire sans conversions avec le logiciel Paraview qui est déjà standard dans le monde de la CFD.

Cependant, son apparente complexité peut rebuter certains utilisateurs, qui voudraient l'utiliser. En effet, ne disposant pas d'interface graphique mais fonctionnant avec un système de fichiers, il nécessite un véritable savoir-faire et une véritable motivation de la part des utilisateurs pour l'utiliser sans problèmes. C'est pourquoi, en tant qu'élèves ingénieurs dans l'aéronautique et le spatial, il nous a paru bon de nous pencher sur cet outil qui ne peut qu'intéresser les entreprises du fait de sa gratuité. L'objectif de ce PFE est donc double : D'une part, déterminer la pertinence d'utilisation d'OpenFOAM face à des logiciels commerciaux ayant déjà fait leurs preuves dans le cadre de l'aérodynamique à travers des tests et des comparaisons ; et d'autre part donner les clefs permettant au lecteur de prendre en main de façon rapide et intuitive cet outil à travers des explications didactiques de ses aspects les plus fondamentaux.

Cette étude peut intéresser également d'autres secteurs que l'aérodynamique, puisqu'OpenFOAM permet en outre de faire des études de combustion, de mécanique, d'électromagnétique ou de finance et que, pour chacune de ces études, les bases de l'utilisation d'OpenFOAM sont les mêmes. Ce projet se limite à l'utilisation d'OpenFOAM du point de vue du non-programmeur et ne rentre pas dans le détail de la création de solveur, mais uniquement de l'utilisation des solveurs existants.

Il se compose de cinq grandes parties dont :

- Une présentation générale détaillant les outils spécifiques à OpenFOAM.
- Une simulation de base d'un écoulement potentiel autour d'un cylindre.
- Une simulation d'un écoulement autour d'un profil d'aile avec validation des résultats.
- Une simulation d'un écoulement autour d'un avion.
- Et enfin, une partie présentant un écoulement supersonique dans un divergeant.

Cette étude s'est appuyée sur l'utilisation d'OpenFOAM 2.3.0 dans la version 14.10 d'Ubuntu 64 bit en natif ainsi que par le biais de la machine virtuelle Oracle VirtualBox. La Machine utilisée est un Laptop core deux duo Intel® Pentium(R) CPU 2020M @ 2.40GHz x 2 avec 1.8 GiB de mémoire vive.

CHAPITRE 1

CHAPITRE 1

PRESENTATION D'OPENFOAM

1.1. Introduction

OpenFOAM (Open Field Operation And Manipulation) est un solveur multi-physique (mécanique des solides, électromagnétisme, finance, ...etc.) orienté surtout vers la mécanique des fluides. Il a été conçu en 1989 à l'Imperial College London, et est utilisé pour résoudre les équations aux dérivées partielles par la méthode des volumes finis.

Il s'agit d'un code open-source développé en C++ (programmation orienté objet), conçu comme une boîte à outils, il contient plus de 200 programmes (des préprocesseurs, des solveurs et des outils de post-traitement), il est réutilisable et modifiable à souhait, et l'utilisateur peut par ce biais programmer un nouveau solveur correspondant à son cas d'étude. Une autre caractéristique du logiciel est qu'il ne possède pas une interface graphique (comme Fluent, Code_Saturne, ...etc.), toute la simulation numérique et le conditionnement du cas se fait dans des fichiers textes.

Ce code de calcul est une solution open source, et donc permet d'éviter les codes de calcul payants avec un coût exorbitant des licences. C'est un outil puissant, personnalisable, et offre à l'utilisateur une grande liberté d'accès à l'intérieur du code, comprendre tout le fonctionnement du code (contrairement à Fluent par exemple où on ne sait jamais ce qui est à l'intérieur), et créer son propre solveur et ses propres modèles qui correspondent le mieux avec son cas d'étude. De même, ce logiciel est assez puissant en termes de performance, il permet de lancer des calculs sur un grand nombre de cœurs sans surcoût.

Ce logiciel présente une multitude de solveurs relatifs à différents domaines d'application, tels que les écoulements compressibles, les milieux multiphasiques, réactifs, ...etc. OpenFOAM contient aussi ce qu'on appelle des "utilités standards", qui ont pour rôles la génération du maillage (BlockMesh, ExtrudeMesh, ...), la conversion et la manipulation du maillage (AnsysToFoam, Fluent3D MeshToFoam, ...), ainsi que d'autres outils pour le calcul et le post-traitement (Post-processing graphics, Post-processing data converters, ...).

Pour télécharger le logiciel, il suffit d'aller au site suivant : [http://www.openfoam.org/download/\[1\]](http://www.openfoam.org/download/[1]) et suivre la démarche qui est bien détaillée (voir **A1**). Vous pouvez trouver sur ce site, des tutoriaux bien détaillés pour se familiariser avec le logiciel, comme vous pourrez aller sur la section "forum" où l'on peut poser des questions en relations avec le logiciel et où vous pourrez recevoir de l'aide et des conseils au cas où vous rencontrez des problèmes.

1.1.1. Histoire

OpenFOAM (Open Field Operation and Manipulation[2]) est une boîte à outils de simulation multi-physiques principalement axé sur la résolution des équations de la mécanique des fluides. Il est distribué depuis 2004 sous licence open source GNU/GPL par la société britannique OpenCFD Ltd (acquise par SGI le 8 août 2011, puis par ESI Group le 11 septembre 2012). Son développement, en C++, a été amorcé par l'Imperial College London qui souhaitait un code de calcul basé sur la méthode des volumes finis et qui bénéficierait des dernières innovations en termes de langage informatique. La dernière version 2.4.0 d'OpenFOAM est sortie en Mai 2015.

1.1.2. Domaines d'applications

OpenFOAM, comme tout logiciel de CFD, est utilisé dans plusieurs domaines de la recherche scientifique, telle que la médecine, l'architecture, l'aéronautique, la construction d'automobiles, ou encore dans la finance. Beaucoup d'entreprises ont choisi ce logiciel à cause de sa souplesse et son adaptation facile.

1.2. Quelques définitions

- **Un logiciel open source** : « Open source[3] » désigne un logiciel dans lequel le code source est à la disposition du grand public. C'est généralement un effort de collaboration qui permet à des programmeurs d'améliorer ensemble le code source. L'open source a déjà enveloppé tous les grands domaines du système d'information des administrations mondiales: environnements, médecine, outils d'ingénierie, solutions de réseaux et sécurité, ...etc. Les logiciels open source sont désormais au

même rang que les logiciels commerciaux du point de vue du secteur public. Les décideurs effectuent d'ailleurs de plus en plus leur choix à partir d'un jugement éclairé, en comparant systématiquement les logiciels propriétaires aux logiciels libres.

- **Un logiciel orienté objet :** Il consiste en la définition et l'interaction de petits programmes appelés objets[4]. Un objet représente une idée ou un concept qui possède une structure interne et un comportement, et sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations, l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues.
- **Une licence GPL:** La licence publique générale GNU[5], ou GNU General Public License, voire simplement « GPL », est une licence qui fixe les conditions légales de distribution des logiciels libres du projet GNU. La GPL met en œuvre la notion de « Copyleft », un jeu de mots anglais faisant référence à la notion de copyright. Le copyright garantit exclusivement les droits de l'auteur, le « Copyleft » s'attarde tout particulièrement aux droits des utilisateurs, et vise à préserver la liberté d'utiliser, d'étudier, de modifier et de diffuser le logiciel et ses versions dérivées.
- **Shell de Linux :** Le Shell[6] est un langage de programmation interprété autorisant la récursivité mais c'est avant tout l'interpréteur de la commande standard du système Linux. Linux a une variété de Shell, le Bourne shell (sh) le Korn shell (ksh) et, le plus populaire, le Bourne Again shell (bash). Un programme en Shell automatise un grand nombre des commandes, il peut aussi faire des boucles afin de répéter un travail qui a, par exemple, le même algorithme mais avec différentes données à chaque fois.

1.3. Les fonctionnalités d'OpenFOAM

Le logiciel est considéré comme une boîte à outils CFD, avec 200 programmes totalement accessibles, l'utilisateur à l'avantage de modifier ou créer ses propres solveurs qui conviennent au cas étudié.

- **Mailler** : OpenFOAM permet de faire des maillages entièrement structurés par l'utilitaire « blockMesh », ou de type non structuré par « snappyHexMesh » qui est un outil qui permet, à partir d'une simple géométrie 3D, de mailler le domaine qui l'entoure en quelques opérations seulement. OpenFOAM permet de travailler à partir de maillages faits depuis d'autres logiciels comme Gambit, importés sous des formats comme .neu ou .msh. Enfin, OpenFOAM est aussi capable de faire des opérations de transformation du maillage comme des rotations ou des déformations.
- **Faire de la CFD** : OpenFOAM est donc capable de résoudre les équations de la mécanique des fluides via la méthode des volumes finis, pour des écoulements stationnaires, non stationnaires, compressibles, incompressibles, mettant en jeu de la convection, de la combustion, plusieurs phases, etc. OpenFOAM permet également de faire des études mécaniques des structures.
- **Créer ses propres solveurs** : OpenFOAM permet en outre de créer ses propres solveurs de modifier un solveur existant. Ainsi, cet outil n'a aucune limite et permet a priori de tout résoudre, selon la motivation et le degré de compétence de l'utilisateur.
- **Post-traiter** : Bien qu'OpenFOAM ne bénéficie pas d'outil graphique permettant de post-traiter les résultats de calcul, il a été programmé pour permettre une utilisation très simple du logiciel de post-traitement open source Paraview. Il permet aussi d'exporter les résultats vers d'autres logiciels de post-traitement. En plus, il admet de créer des graphiques sous des logiciels comme Gnuplot.

1.4. Les avantages et les inconvénients d'OpenFOAM

OpenFOAM a beaucoup satisfait ses utilisateurs, peut-être parce qu'ils ne sont pas en position de juger un travail qui est tout à fait libre et gratuit. Cependant il est important de ne pas oublier qu'il faut continuer à perfectionner ce logiciel.

1.4.1. Les Avantages

Les points forts d'un logiciel open source sont très loin de ce que beaucoup de gens pensent. La communauté des développeurs est très active, et l'utilisateur n'a pas besoin d'attendre pour recevoir les nouveautés d'OpenFOAM, sans parler de l'aide qu'on peut avoir sur n'importe quel sujet à n'importe quel moment.

- a- La gratuité : Le principal avantage d'OpenFOAM est bien entendu sa gratuité, qui n'est pas négligeable devant le coût excessif des licences de logiciels de CFD comme Fluent ou Star-CCM+.
- b- La puissance : OpenFOAM est également un outil puissant qui contient des algorithmes intelligents permettant de réduire les temps de calcul en fonctionnant sous forme de pas de temps dans lequel chaque grandeur a un nombre propre d'itérations. Ainsi, dans un même pas de temps, l'équation de continuité selon x peut être itérée 2 fois et celle selon y 10 fois.
- c- Accès aux sources : OpenFOAM n'est pas une "boîte noire", et on peut programmer ou modifier des programmes qui existent selon notre désir.
- d- Facilité de programmer des équations : L'équation de Navier-Stokes peut s'écrire sous la forme :

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p \quad (1.1)$$

Le schéma équivalent de l'équation est :

```

solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);

```

1.4.2. Les inconvénients d'OpenFOAM

Malgré la grande revendication et l'accroissement de la communauté des développeurs, OpenFOAM a n'a pas plut aux utilisateurs néophytes et débutants. Certaines solutions ont été mis en place comme les émulations sous Windows ou encore les interfaces graphique (Discretizer et Discretizer setup), mais ils été rapidement mis à l' écart.

- a- *La complexité*** : Le principal défaut d'OpenFOAM est sans aucun doute sa grande complexité qui peut repousser aux premiers abords les personnes inexpérimentées et nécessite un temps pour la prise en main. Contrairement aux autres logiciels de CFD payants, OpenFOAM ne bénéficie pas d'une interface graphique intuitive, bien que quelques outils annexes soient en cours de développement comme « Discretizer » et « Discretizer Setup », mais fonctionne avec un système de fichier difficile à prendre en main et recèle de nombreuses subtilités qui ne peuvent être appréhendées que par le biais de l'expérience.
- b- *Le système d'exploitation*** : Malgré que pour un programmeur expert cela peut être un avantage, OpenFOAM étant sous Linux, il peut être long pour un débutant de prendre en main ce système d'exploitation. En effet, bien qu'il existe une émulation d'OpenFOAM sous Windows, facilement installable, permettant à l'utilisateur de retrouver les mêmes commandes qu'avec une console Linux dans une console Msys ou Cygwin et permettant en outre d'utiliser Discretizer et DiscretizerSetup, cette solution non optimum, sujette à des conflits entre les fichiers émulés et les fichiers Windows. L'utilisation sous Linux est, sur le long terme, bien plus simple et permet surtout de ne pas dépendre d'une compagnie privée pour avoir la dernière version du logiciel. Il peut être aussi contraignant de jongler entre le système d'exploitation Linux et Windows dans le cas où l'utilisateur voudrait utiliser sur le même poste des logiciels comme ANSYS et OpenFOAM.

Note : On peut tout de même noter que Linux peut être implémenté dans Windows ou Mac OS via une machine virtuelle. En effet les machines virtuelles

sont des programmes qui permettent de lancer un système d'exploitation différent de celui en cours de la même façon qu'on lance un petit programme. L'intérêt de cette option est que l'on peut profiter d'OpenFOAM en natif sans avoir à booter sous Linux. Cependant, la machine virtuelle consomme inévitablement des ressources processeur, ce qui augmente les temps de calcul.

1.5. La logique d'OpenFOAM

Avec OpenFOAM, l'utilisateur n'a pas affaire à une interface graphique (sauf avec Discretizer et Discretizer Setup) avec des menus et des boîtes de dialogue comme avec un logiciel de CFD ou de maillage classique, mais à un système de fichiers. La connaissance de ce système de fichier est fondamentale pour toute personne désireuse d'utiliser OpenFOAM. Le logiciel graphique Paraview permet seulement de visualiser les résultats de calculs en allant lire ces fichiers. Le phénomène étudié avec OpenFOAM (c'est-à-dire un maillage et un calcul CFD) est caractérisé par son dossier que l'on nomme ici un « Dossier de cas ». La structure de ce dossier est donnée sous forme d'arbre dans la figure ci dessous. Cette structure n'est pas fixe et est susceptible d'être étendue pendant l'utilisation. Néanmoins, la base est toujours la même.

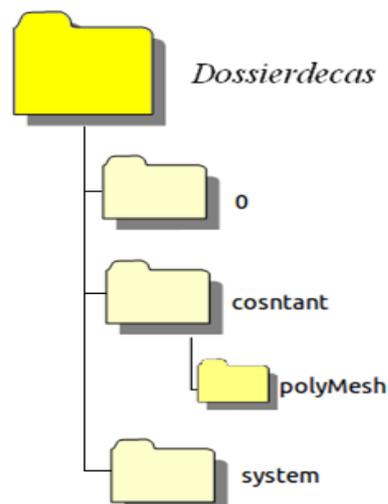


Figure 1.1 :Structure basique d'un dossier de cas sur OpenFOAM.

Note : Ne pas utiliser des espaces dans le nom d'un dossier de cas. Ex :
« *Dossierdecas* ».

1.5.1. Les dossiers temporels « 0 »

Dans le dossier de travail, les dossiers temporels sont ceux qui contiennent les valeurs des différentes grandeurs calculées au cours du temps (pression, vitesse, intensité turbulence, etc.) réunies dans des fichiers dont le nom correspond à ces grandeurs (ex « p » pour la pression, « U » pour la vitesse, « k » pour l'énergie cinétique turbulente, etc.). Le nom de ces dossiers temporels correspond au temps en secondes associé à ces valeurs (ex « 0 », « 1 », « 0.1253 », etc.). C'est dans le premier dossier temporel que l'utilisateur spécifie les conditions initiales et les conditions aux limites, afin que le solveur d'OpenFOAM « s'accroche » à elles pour commencer ses calculs et créer d'autres dossiers temporels avec les valeurs calculées par le solveur.

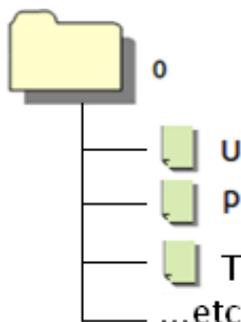


Figure 1.2 : Exemple d'dossier temporelle « 0 »

1.5.2. Le dossier « constant »

Le dossier constant contient les fichiers utilisateurs liés aux propriétés indépendantes du temps telles que la viscosité du fluide dans « transportProperties », le modèle de turbulence de l'écoulement dans « turbulenceProperties » ou d'autres fichiers qui dépendent du type d'écoulement à résoudre (non visqueux, turbulent, etc.).

En outre, le dossier « constant » contient un sous-dossier « polyMesh » qui contient les fichiers machines relatifs au maillage : « boundary », « faces », « neighbour », « owner », « points », ainsi que le fichier utilisateur « blockMeshDict » si le maillage est généré par blockMesh, sinon, ça signifie que le maillage est obtenu autrement (ex: par convertissement).

Le dossier « constant » peut également contenir un dossier « triSurface » avec des fichiers qui contiennent les géométries de surfaces de type STL ou OBJ dans le cas d'un maillage via la commande «snappyHexMesh».

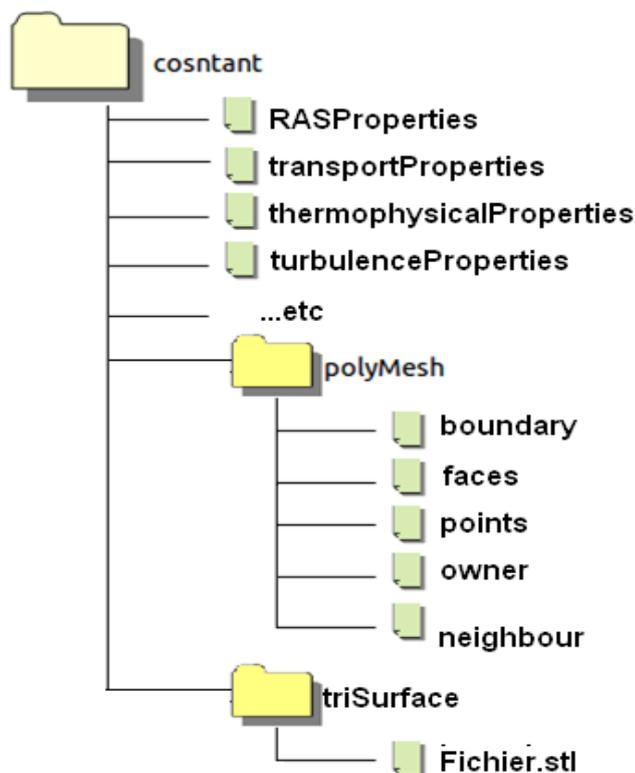


Figure 1.3 : Exemple d'un dossier « constant »

1.5.3. Le dossier « system »

Le dossier « system » est un dossier fondamental qui contient un fichier utilisateur « controlDict » dans lequel celui-ci spécifie le nom du solveur utilisé et les contraintes temporelles du calcul (temps initial, temps final, intervalle de temps entre deux calculs, intervalle entre deux dossiers temporels d'écriture, etc.). Le dossier « system » contient également deux fichiers utilisateurs « fvSchemes » et « fvSolution » dans lequel l'utilisateur paramètre les algorithmes de résolution utilisés. Ce dossier peut contenir aussi le fichier « snappyHexMeshDict » si l'utilisateur veut faire un maillage non structuré autour d'un objet, et le fichier « decomposeParDict » qui permet de décomposer le problème par processeur afin d'accélérer l'exécution.

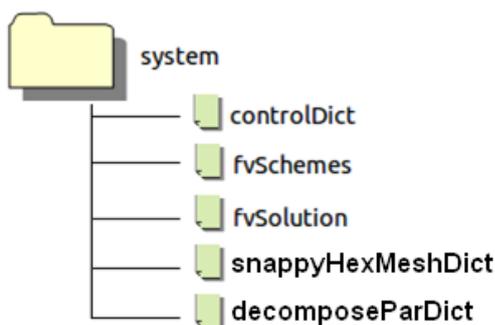


Figure 1.4 : Exemple d'un dossier « system »

1.5.4. Les fichiers d'OpenFOAM

Les fichiers sous OpenFOAM se présentent tous de la même manière. Un entête, une ligne d'ouverture, un corps et une ligne de fermeture. On peut aussi trouver des lignes écrites en C++, mais en générale, chaque fichier représente le nombre des entrées nécessaires pour le programme, et à condition que la syntaxe soit correcte, ces entrées peuvent être écrites manuellement, généré par un autre compilateur (la fonction « open » dans le FORTRAN) ou tout simplement écrites en C++. L'exemple suivant montre une partie d'un fichier blockMeshDict. Le fichier peut être trouvé dans:

Tutorials/basic/potentialFoam/cylinder/constant/polyMesh/blockMeshDict

- **L'entête** : C'est la partie qui contient les informations de base sur le fichier. La première partie en bleu est la même dans tous les fichiers d'OpenFOAM, c'est une signature du groupe des auteurs du logiciel. La version spécifique, comme indique son nom, la version du fichier, puis le format, qui est en général ascii, ensuite la classification du fichier « class », c'est-à-dire ça catégorie, qui peut être un dictionnaire, un paramètre scalaire ou vectoriel, ou autre entité physique ou bien verbale. Cette dernière ligne est suivie par la notion « object » qui signifie « objet ». Les langages de programmation qui sont orientés objets, comme le C++, utilisent les notions « class » et « objet » pour déclarer des types et des opérations associés qui font parties des langages verbal et mathématique utilisé dans les sciences d'ingénieur.


```

convertToMeters 1;
vertices #codeStream
{
  codeInclude
  #{
    #include "pointField.H"
  #};

  code
  #{
    pointField points(19);
    points[0] = point(0.5, 0, -0.5);
    points[1] = point(1, 0, -0.5);
    points[2] = point(2, 0, -0.5);
    points[3] = point(2, 0.707107, -0.5);
    points[4] = point(0.707107, 0.707107, -0.5);
    points[5] = point(0.353553, 0.353553, -0.5);
    points[6] = point(2, 2, -0.5);
    points[7] = point(0.707107, 2, -0.5);
  #};
}

```

Figure 1.7 : Exemple d'un corps de fichier blockMeshDict associé à un programme en C++.

- **Ligne de fermeture:** C'est la dernière ligne dans le fichier. Elle commence par un double slash qui signifie en C++ le début d'un commentaire. Cette ligne peut être supprimée sans affecter le reste du programme.

```
// ***** //
```

Figure 1.8 : Ligne de fermeture d'un fichier sur OpenFOAM.

1.6. Les outils d'OpenFOAM

OpenFOAM est par définition une boîte à outils CFD, Plus de 200 utilitaires pour différentes étapes de simulation. Grâce à ses programmes de conversions, le logiciel peut être également utilisé comme intermédiaire pour exporter ou importer différents résultats.

1.6.1. Pré-procession

Comme on l'a déjà mentionné, OpenFOAM admet plusieurs méthodes pour la génération et la manipulation de maillage. Il peut faire des maillages entièrement structurés comme il peut faire des maillages non structurés autour d'une géométrie importé par SolidWorks.

A- BlockMesh

La commande « blockMesh », sous OpenFOAM, permet de générer un maillage 3D hexagonal de type structuré à partir d'un seul fichier nommé « blockMeshDict » placé dans le dossier « constant/polyMesh » du dossier de travail. Un maillage de type « blockMesh » est un maillage structuré fait à partir d'un ou plusieurs blocs géométriques juxtaposés les uns aux autres. Chaque bloc est défini par 8 sommets, certains sommets peuvent être confondus, l'utilisateur peut spécifier la nature de la liaison entre les points (« spline », « polyLine », « arc », « line (par défaut) »), le nombre de mailles dans les trois directions et les déformations facultatives de ses arêtes. L'avantage de ce genre de maillage est qu'il est toujours structuré, quel que soit la forme des blocs qui le composent. Par contre, l'inconvénient de ce genre de maillage est qu'il peut demander un travail énorme pour mailler des géométries pourtant relativement simples.

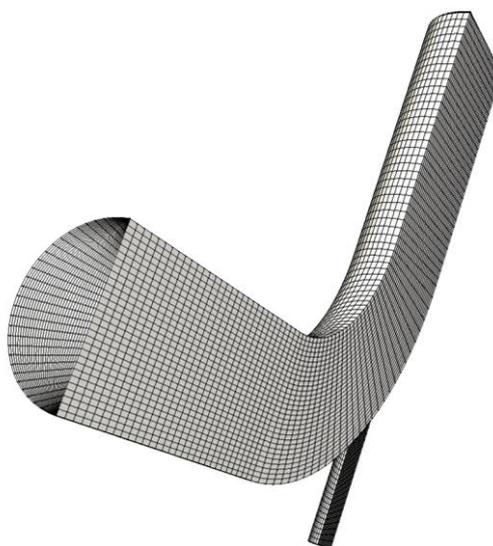


Figure 1.9 : Maillage fait par blockMesh.

Le fichier doit contenir les lignes suivantes :

- **ConvertToMeters** : C'est l'échelle par laquelle les points sont écrits. Le produit de ce facteur avec x,y et z donne les vrais coordonnées en mètre.

```
convertToMeters    1;
```

- **Vertices** : contiennent les points nécessaire pour créer un bloc, leur classement est très importants car chaque points est connue par une étiquette (0, 1, 2,3, ...etc.) qui indique son emplacement.

```
vertices
(
(1 0 0)           //le point 0
(1 1 0)           // le point 1 ...
);
```

- **Blocks** : Contiennent les blocs de la géométrie 3D. Un bloc est définie par 8 points ordonné de la façon suivante : (0 1 2 3 4 5 6 7) ou bien (0 1 5 4 3 2 6 7) ou (3 0 4 7 2 1 5 6) ...

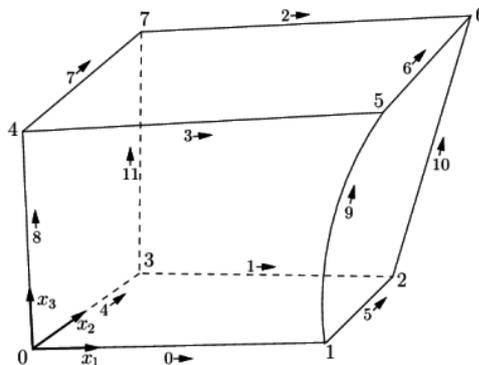


Figure 1.10 : Logique de la génération des blocs sur blockMesh.

```

blocks
(
hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1) //le block 0
);

```

Où : (20 20 1) représente le nombre des nœuds selon x_1 , x_2 et x_3 .

simpleGrading (1 1 1) définit le rapport : $\frac{\text{la distance entre les deux premiers nœuds}}{\text{la distance entre les deux derniers nœuds}}$

Note : Pour raffiner dans deux directions opposées, il est nécessaire de créer au moins deux blocs. Par exemple :

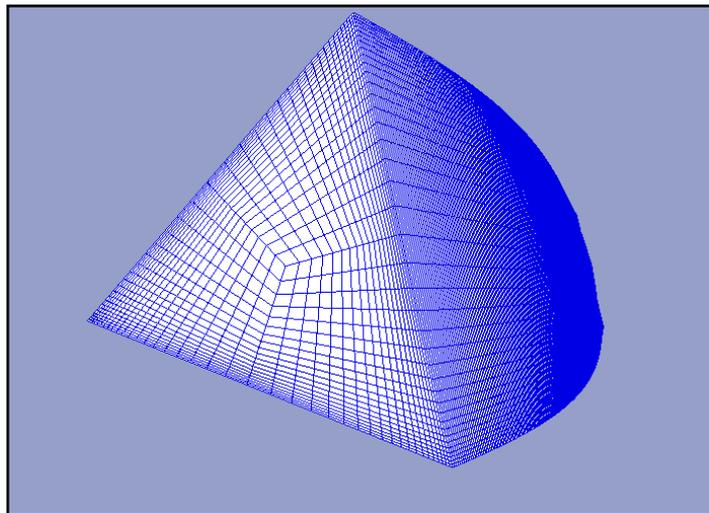


Figure 1.11 : Tuyère triangulaire composée de 3×100 blocs quadrilatéral avec raffinement des parois.

- **Edges** : Chaque paire de points est liée par une ligne droite par défaut, mais l'utilisateur peut définir une trajectoire dans une liste des points.

```

edges
(
arc 1 2 (0.70710,0.7071 0) // Un arc de 1 a 2 passant par le point indiqué.
spline 0 1 (//Des interpolations entre les deux points 0 et 1 passant par la liste suivante.
( 0.00000000 0.705229998 0.630990028 )
( 0.00000000 0.705229998 0.556760013 )
( 0.00000000 0.705229998 0.482520014 )
)
);

```

boundary : Contiennent les noms des régions (libre à l'utilisateur) ainsi que leur nature.

```

boundary
(
Inlet // Le nom de la région, la majuscule est un tout autre caractère.
{
type patch; // La nature, ou le type de la région.
faces
(
(0 10 17 7) // Les faces qui la constitue.
(7 17 16 6)
(6 16 15 5)
);
}
airfoil// Il est préférable de ne pas utilisé les mot de la machine comme « wall », « patch ».
{
typewall;
faces
(
(8 18 13 3)
(9 19 18 8)
(3 13 19 9)
);
}
\..

```

mergePatchPairs : Un maillage peut être créé par plusieurs blocs. Une connexion entre ces blocs est nécessaire, pour cela, il existe deux possibilités :

- « face matching » : Qui réunit une face d'un bloc avec une autre d'un autre bloc, à condition qu'ils aient les mêmes points de constitution. Cette option est appliquée par défaut si l'utilisateur n'indique pas autrement.
- « face merging » : Un groupe de faces d'un bloc est connecté avec un autre groupe de faces d'un autre bloc pour définir des faces interne connectant les deux blocs.

```
mergePatchPairs
(
// Si vide, les faces internes sont regroupés automatiquement par « face matching ».
);
```

B. SnappyHexMesh

La commande « snappyHexMesh » permet de modifier un maillage OpenFOAM existant (typiquement, un maillage de type « blockMesh ») à l'aide d'une géométrie surfacique en trois dimensions sous format STL. Pour mailler un domaine autour d'une géométrie quelconque, par exemple un avion, il suffit d'avoir sa géométrie sous format STL, de créer un simple maillage de type blockMesh représentant le domaine autour de l'avion et de « demander » à snappyHexMesh de faire le reste de façon tout à fait automatique, comme le montre la figure suivante :

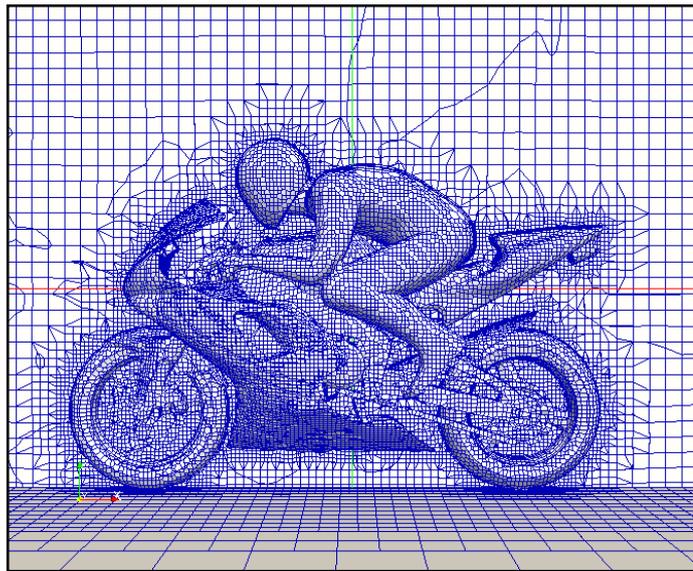


Figure 1.12 :Un maillage autour d'unemoto avec snappyHexMesh.

Pour lancer l'utilitaire snappyHexMesh l'utilisateur doit fournir :

- Un fichier sous format STL de la géométrie dans le répertoire constant/triSurface.
- Un fichier blockMeshDict dans le dossier constant/polyMesh qui définit le domaine extérieur.
- Un fichier snappyHexMeshDict dans le répertoire **system/** avec les données nécessaires :

```

geometry
{// Liste des géométries STL et zones de raffinement.};
castellatedMeshControls
{// Paramètres du raffinement.}
snapControls
{// Paramètres de la découpe des mailles pour épouser la géométrie.}
addLayersControls
{// Paramètres des raffinements des couches limites.}
meshQualityControls
{// Paramètres propres au maillage dans son ensemble.}
debug 0; // Rapport des erreurs
mergeTolerance 1E-6; // Niveau de tolérance du maillage.

```

C- Conversions de maillage

OpenFOAM contient plusieurs convertisseurs de maillage ce qui facilite la comparaison des résultats entre les logiciels de CFD. Il peut importer et exporter des maillages de différentes géométries. On cite les plus connues :

Tableau 1.1 : Les commandes de conversions de maillage[7].

ansysToFoam	Convertir un maillage fait par ANSYS vers OpenFOAM (.ans).
cfx4ToFoam	Convertir un maillage fait par CFX 4 vers OpenFOAM.
datToFoam	Lire un fichier .dat et le convertir en fichier de points utilisé avec blockMesh.
fluentMeshToFoam	Convertir un maillage fait par Fluent vers OpenFOAM.
foamMeshToFluent	Convertir un maillage fait par OpenFOAM vers Fluent.
foamToStarMesh	Convertir un maillage fait par OpenFOAM vers PROSTAR.
foamToSurface	Convertir un maillage fait par OpenFOAM vers un fichier .STL
gambitToFoam	convertir un maillage fait par GAMBIT vers OpenFOAM.
gmshToFoam	convertir un maillage fait par Gmsh vers OpenFOAM.
ideasUnvToFoam	convertir un maillage .unv vers OpenFOAM.
mshToFoam	convertir un maillage .msh vers OpenFOAM.
star4ToFoam	convertir un maillage fait par STAR-CD (v4) PROSTAR vers OpenFOAM.

1.6.2. Les solveurs

La notion de solveur est fondamentale en ce qui concerne OpenFOAM. En effet, la première chose à se demander, lors de l'étude d'un cas avec OpenFOAM, est celle du solveur à utiliser. Contrairement à Fluent, OpenFOAM a, du point de vue utilisateur, presque autant de solveurs différents que de type de cas d'étude possible. Par exemple, il a un solveur nommé « icoFoam » destiné uniquement aux écoulements incompressibles laminaires non stationnaires. Un autre solveur nommé « simpleFoam » est destiné aux écoulements incompressibles turbulents stationnaires. Il n'est pas nécessaire de choisir les équations à résoudre, comme pour Fluent, il suffit uniquement de choisir le bon solveur. Le tableau 1.2 présente les solveurs qui peuvent être utiles dans le cadre de la CFD.

Tableau 1.2 : Les solveurs disponibles sur OpenFOAM[8].

Nom des solveurs		Type d'écoulements
Basique	laplacianFoam	Diffusion thermique dans un solide.
	potentielFoam	Pour initialiser un écoulement potentiel simple avant la résolution des équations de Navier-Stockes.
Incompressibles	boundaryFoam	Écoulement permanent 1D avec turbulences.
	channelFoam	Pour un écoulement dans un tube avec de grandes turbulences.
	icoFoam	Transitoire, Laminaire et isotherme pour fluide Newtonien.
	nonNewtonianIcoFoam	Transitoire, laminaire, isotherme, non-Newtonien.
	pimpleDyFoam	Transitoire, fluide Newtonien sur parois en translation.
	pimpleFoam	Transitoire, grand pas de temps.
	pisoFoam	Transitoire.
	shallowWaterFoam	Transitoire, fluide non visqueux et peu profond, rotation.
	simpleFoam	Permanent, turbulent.

Compressibles	rhoCentralFoam	Solveurs basé sur l'algorithme de Kurganov&Tadmor.
	rhoPimpleFoam	Laminaire ou turbulent pour chauffage, climatisation et ventilation.
	rhoPisoFoam	Non permanent, laminaire ou turbulent.
	rhoPorousMRFPimpleFoam	Géométrie poreuse et MRF, chauffage, climatisation, ventilation, laminaire ou turbulent.
	rhoPorousSimpleFoam	Fluide permanent avec turbulence RANS et traitement poreux géométrie.
	rhoSimpleFoam	Permanent, SIMPLE pour laminaire ou turbulent RANS.
	sonicDyFoam	Maillage en mouvement, transitoire, gaz transsonique/supersonique, laminaire ou turbulent.
	sonicFoam	Transitoire, gaz transsonique/supersonique, laminaire ou turbulent.
	sonicLiquidFoam	Transitoire, transsonique/supersonique, laminaire, fluide liquide,
Multi-phase	bubbleFoam	2 fluides incompressibles non miscibles (ex : eau gazeuse)
	cavitatingFoam	Transitoire, cavitation.
	compressiblesInterFoam	2 fluides compressibles, isothermes.
	interFoam	2 fluides incompressibles, isothermes.
	interMixingFoam	3 fluides incompressibles dont 2 miscibles.
	interPhaseChangeFoam	2 fluides incompressibles isothermes avec échange de phase.
	multiphaseInterFoam	n fluides incompressibles.
	twoLiquidMixingFoam	2 fluides incompressibles dont une sous forme de gaz ou de bulles dispersés.
	twoPhaseEulerFoam	2 fluides incompressibles non miscibles (ex : eau gazeuse)

1.6.3. Post-traitement « Paraview »

Le post-traitement est l'une des étapes les plus importantes dans une simulation numérique. OpenFOAM présente une immense diversité de choix, des possibilités d'ajouter des fonctions afin de calculer, par exemple des intégrales (coefficient de portance ou de traînée), et de les enregistrer sous différents formats (jPlot, gnuplot, Grace/xmgr, VTK ASCII, ASCII STL, .dat, ...etc) c'est là où le logiciel de post-traitement entre en jeu.

L'utilisation d'OpenFOAM nécessite un système de fichier avec lequel l'utilisateur interagit principalement via une console Linux afin d'exécuter des commandes. Une fois les premiers résultats donnés par le solveur, l'utilisateur utilise Paraview[9] afin de les post-traiter. Paraview étant le logiciel privilégié pour le post-traitement des résultats OpenFOAM, il est indispensable de comprendre la logique de son fonctionnement pour pouvoir utiliser OpenFOAM. Ce qui suit traite de cette logique.

A ce sujet, il faut d'abord savoir que Paraview fonctionne avec un système très particulier de filtres. Après importation d'un paquet de données, l'utilisateur filtre celles qui l'intéressent afin de les afficher à l'écran. Ces filtres peuvent être de multiples sortes. La figure suivante montre un arbre de données. Le paquet principal est le paquet « 3DWing.OpenFOAM ». Sur ce paquet ont été appliqués les filtres « Slice » et « Extract Block » qui ont donné deux nouveaux paquets de données « Slice1 » et « ExtractBlock1 ». Sur le paquet « Slice1 », les filtres « Glyph1 » et « Stream Tracer1 » ont été appliqués, tandis qu'un filtre « Contour1 » a été appliqué au paquet de Donnée « Extractblock1 » :

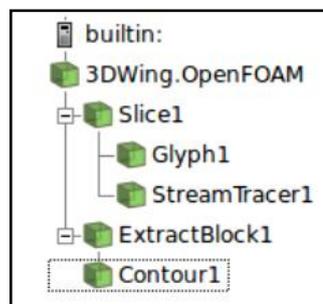


Figure 1.13 : Présentation des filtres sur Paraview

Si quelques filtres clefs se trouvent dans la barre des tâches de Paraview, comme « Slice » pour les plans de coupe, « Glyph » pour les champs de vecteurs et « Stream Tracer » pour les lignes de courant, l'ensemble de ceux-ci se trouvent dans la Barre de menus → Filters → Alphabetical. C'est dans ce menu que l'utilisateur doit chercher pour trouver les filtres dont il a besoin, comme ceux qui servent à faire des symétries, des extractions de données, etc.

- **Importer les données** : Les résultats de calcul OpenFOAM n'étant pas réunis dans un seul fichier mais dans les fichiers comme « U », « p » ou « k » dans des dossiers temporels, il n'est donc pas possible de les lire directement avec Paraview. Cependant, la commande « paraFoam » permet de lancer Paraview et de lire aussitôt les résultats de calcul.

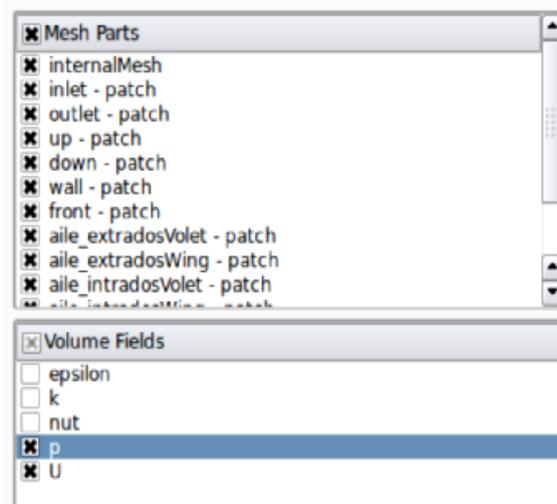


Figure1.14 : Importation du maillage et les paramètres désiré.

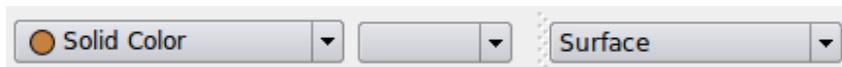
- **Manipuler la fenêtre graphique** : Pour faire pivoter les données affichées dans la fenêtre graphique, utiliser le clic gauche de la souris. Pour les translater, utiliser le clic central (celui de la roulette), et pour zoomer, utiliser soit la roulette soit le clic droit. L'utilisateur peut également faire pivoter la géométrie de façon parallèle au plan de l'écran en utilisant en plus du clic gauche la touche [Shift]. Certains boutons de la barre d'outils permettent, dans l'ordre, de zoomer au mieux, d'user d'une boite de zoom ainsi que de réinitialiser la vue selon les axes x, y ou z :



La barre d'outil permet également, dans l'ordre, de cacher ou non la base (x, y, z) colorée en bas à gauche de la fenêtre graphique, de cacher ou non la position du centre de rotation, de réinitialiser la position de ce centre de rotation ou de le choisir manuellement :



- **Changer l'affichage des données :** La première chose que fait l'utilisateur à ce stade est généralement de changer l'apparence des données affichées. Pour ce faire, il est possible d'utiliser les outils de la barre des tâches afin de changer la coloration (« Solid Color » pour une couleur unie, « p » pour le champ de pression, « U » pour celui de vitesse, etc.) ou le mode de représentation (« Surface » pour surfacique, « Wireframe » pour filaire, etc.) :



1.7. Le dossier « tutorials »

Lors de l'installation d'OpenFOAM on trouve un dossier nommé « tutorials » qui contient des dizaines de dossiers d'exemples qui servent comme modèles pour chaque solveur. Ces dossiers sont classés par catégorie puis par solveur. Il est possible d'accéder directement à ce dossier via la console grâce à la commande « tut ».

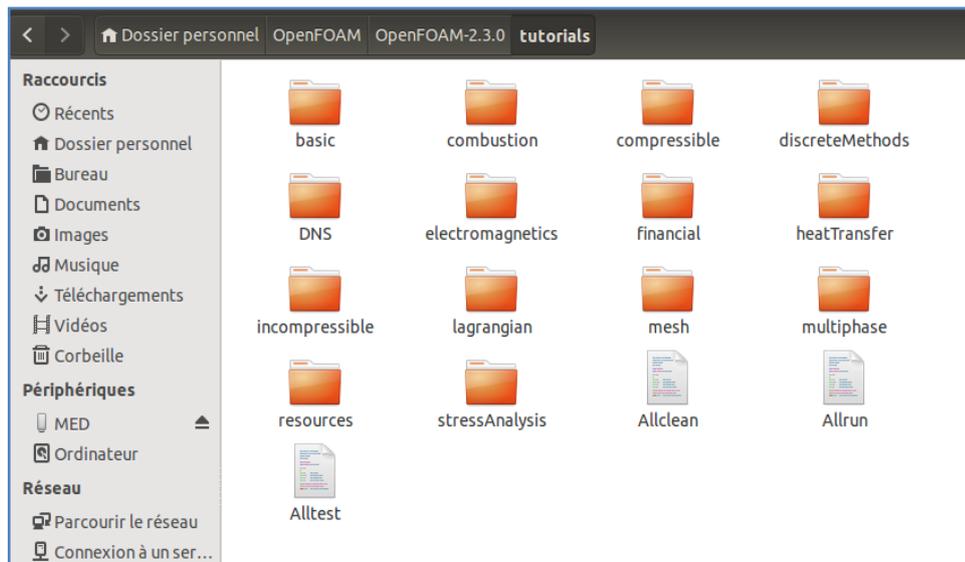


Figure 1.15 :Le dossier « tutorials » sous OpenFOAM-2.3.0.

Chaque tutoriel sert comme exemple sur les outils d'OpenFOAM. On cite à titre d'exemple le dossier « flange ».

1.7.1.Présentation

C'est un exemple simple à partir duquel on peut découvrir des outils de base dans OpenFOAM. Dans cet exemple, la Géométrie et le maillage sont générés avec Ansys. Il s'agit d'une pièce mécanique (un joint de tuyauterie) dont on doit résoudre l'équation de transfert thermique par le solveur « laplacianFoam ».

L'équation de transfert thermique par conduction s'écrit :

$$\frac{\partial T}{\partial t} = \nabla \cdot (D_T \nabla T) \quad (1.2)$$

Avec : D_T coefficient de diffusivité thermique.

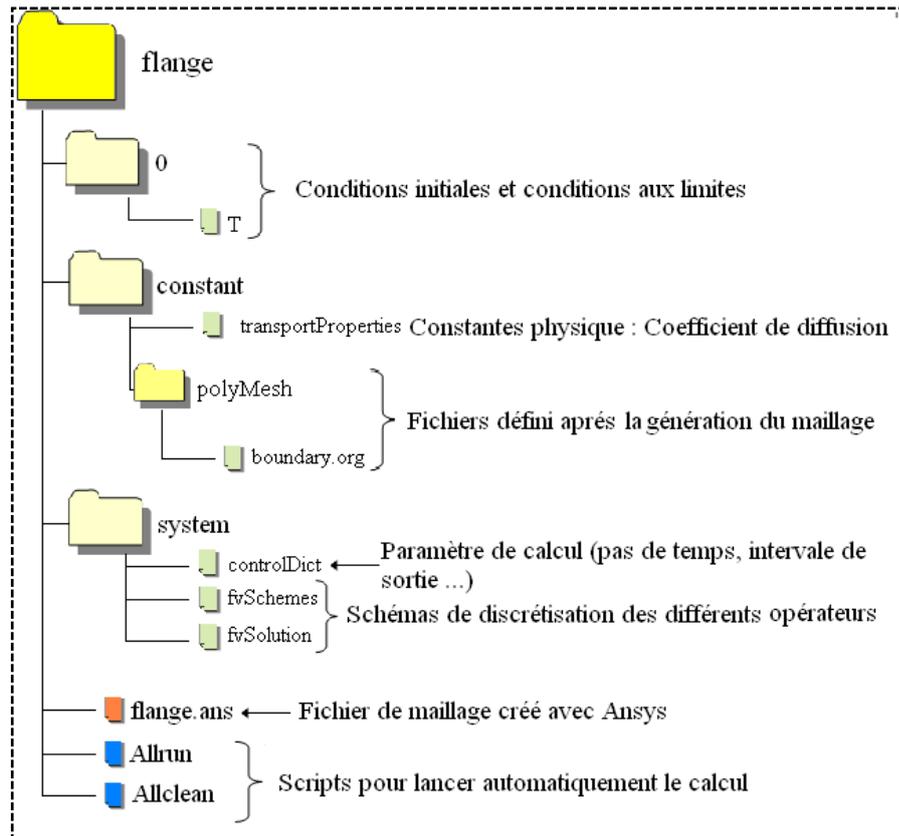


Figure 1.16 : Le système des fichiers du dossier « flange ».

Information sur le système :

Tableau 1.3 : Informations sur le système utilisé.

Système d'exploitation	Ubuntu 14.10
Mémoire	1.8 GiB
Processeur	Intel® Pentium(R) CPU 2020M @ 2.40GHz x 2
Graphiques	Intel® ivybridge Mobile
OS type	64-bit
Disque :	490.1 GB
Version d'OpenFOAM	OpenFOAM v2.3.0

1.7.2. Lancer OpenFOAM dans le Terminal

Pour lancer le Terminal sur Ubuntu il suffit de chercher dans les applications « Terminal » ou simplement appuyé sur « Ctrl + Alt + T ». Une fenêtre apparaîtra avec le nom d'utilisateur :

```
nom@utilisateur:~$
```

Note : Ce nom ne sera pas recopié dans la suite de ce document.

La première commande avant toute utilisation d'OpenFOAM est :

```
of230
```

En peut accéder au dossier « tutorials » directement par la commande raccourci :

```
tut
```

On peut voir que nous avons changé de répertoire ~/OpenFOAM/OpenFOAM-2.3.0/tutorials\$

Pour accéder au dossier « flange » on utilise la commande « cd » suivie de l'emplacement du dossier. Ensuite, on appuie sur « Entrer »

```
cd basic/laplacianFoam/flange/
```

1.7.3. Convertir un maillage

Le dossier contient un fichier de maillage fait par ANSYS « flange.ans ». Il faut d'abords le convertir par une échelle de 0,001.

```
ansysToFoamflange.ans -scale 0.001
```

L'utilitaire ansysToFoam transforme un fichier d'extension « .ans » vers des fichiers accessibles par OpenFOAM. On peut voir directement le résultat de cette commande si on consulte le dossier « constant/polyMesh », le dossier maintenant contient de nouveaux fichiers concernant le maillage converti. L'utilitaire checkMesh vérifie si le maillage est correcte et rapporte les informations de base comme les types de mailles utilisés, son volume maximal et minimal, le nombre des nœuds, ...etc.

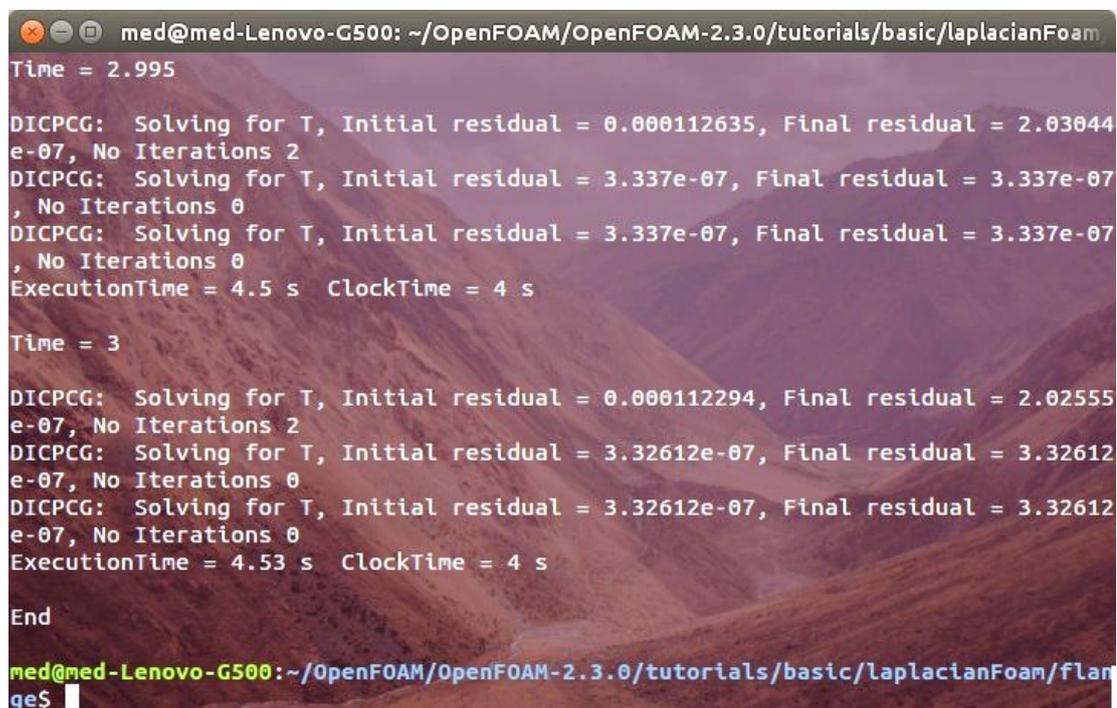
```
checkMesh
```

1.7.4. Lancer le calcul

Le cas étudié est un cas simple de transfert thermique par convection. Le solveur choisi pour ce type de simulation est LaplacianFoam. La commande suivante permettra de lancer le solveur :

```
laplacianFoam
```

Note : l'utilisateur peut écrire seulement « lapl » ensuite appuyer sur « Tab », si c'est la seule commande possible, le mot sera terminer automatiquement sinon il faut appuyer une deuxième fois pour voir les commandes possibles.



```

med@med-Lenovo-G500: ~/OpenFOAM/OpenFOAM-2.3.0/tutorials/basic/laplacianFoam
Time = 2.995
DICPCG: Solving for T, Initial residual = 0.000112635, Final residual = 2.03044
e-07, No Iterations 2
DICPCG: Solving for T, Initial residual = 3.337e-07, Final residual = 3.337e-07
, No Iterations 0
DICPCG: Solving for T, Initial residual = 3.337e-07, Final residual = 3.337e-07
, No Iterations 0
ExecutionTime = 4.5 s   ClockTime = 4 s

Time = 3
DICPCG: Solving for T, Initial residual = 0.000112294, Final residual = 2.02555
e-07, No Iterations 2
DICPCG: Solving for T, Initial residual = 3.32612e-07, Final residual = 3.32612
e-07, No Iterations 0
DICPCG: Solving for T, Initial residual = 3.32612e-07, Final residual = 3.32612
e-07, No Iterations 0
ExecutionTime = 4.53 s   ClockTime = 4 s

End
med@med-Lenovo-G500:~/OpenFOAM/OpenFOAM-2.3.0/tutorials/basic/laplacianFoam/flan
ge$

```

Figure 1.17 :L'affichage des résiduels sur le terminal.

On peut constater que l'exécution a pris 4,53 secondes en temps réel alors que le temps de simulation a été fixé sur 3s. On remarque aussi que 30 nouveaux dossiers sont générés, chacun contient les fichiers : « gradTx, gradTy, gradTz, T » et un dossier « uniform ».

1.7.5. Visualisation des résultats sous Paraview

Pour lancer Paraview il suffit d'écrire dans le terminal la commande suivante :

```
paraFoam
```

- La fenêtre de Paraview s'ouvre. Il faut ensuite s'assurer que les cases des paramètres en question sont cochées, Ensuite, On appuie sur « Apply ».

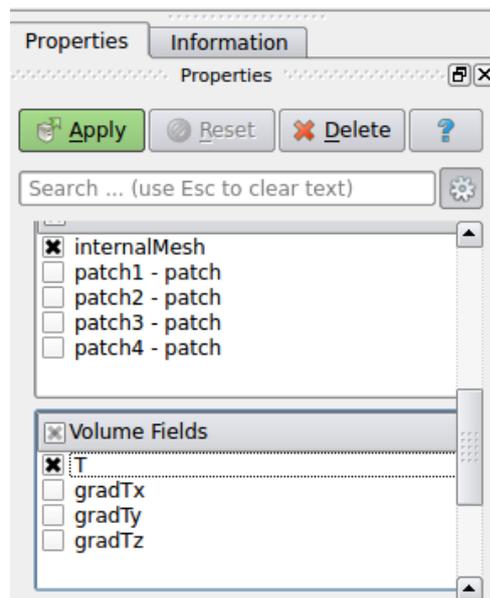


Figure 1.18 : L'onglet « Properties » sur Paraview.

- On choisit « Surface With Edges » pour visualiser le maillage.

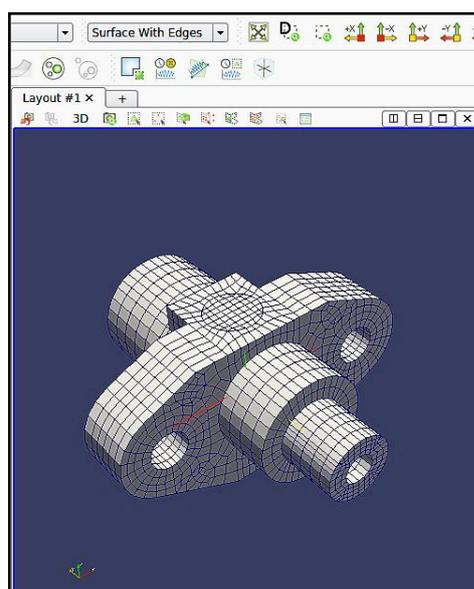
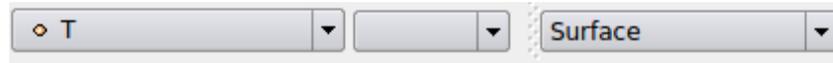


Figure 1.19 : Visualisation du maillage.

- Il faut d'abord choisir les paramètres à visualiser :



- Enfin, on appuie sur « Play » pour voir l'évolution de la température :

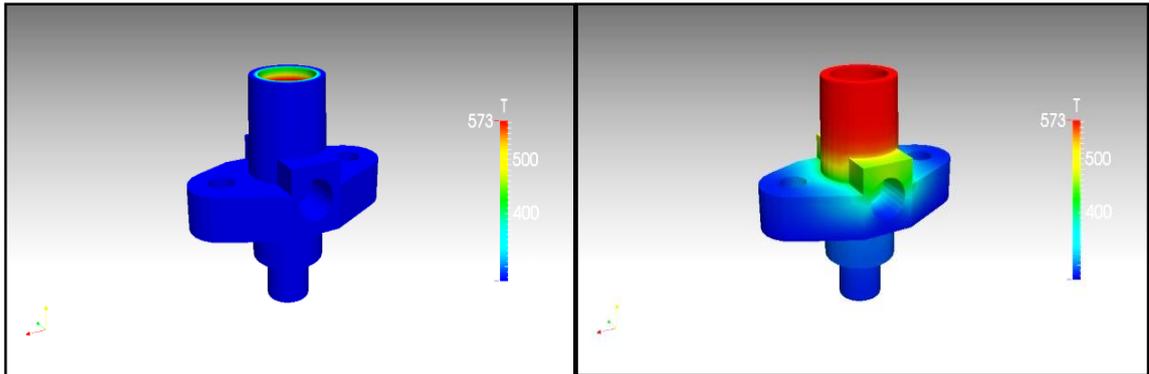


Figure 1.20 : Résultat final de la simulation.

Le résultat semble logique du point de vue physique, comme les conditions aux limites sont imposées fixes (573K en haut et 273K en bas), la température prend des valeurs entre 573K et 273K graduellement.

1.7.6. Exportation des résultats

Dans le tutoriel présent existe des exemples d'exportation des résultats vers des logiciels de post-traitement différents de Paraview. Les deux premiers pour le logiciel Ensign et la troisième, pour le format standard VTK.

```
foamToEnsign
foamToEnsignParts
foamToVTK
```

1.7.7. Auto exécution, les scripts Allrun et Allclean

Dans certains dossiers tutoriels, il existe des scripts « shell scripts », qu'on peut exécuter directement pour donner l'ordre à la machine de faire des exécutions automatiques. Dans la fenêtre Terminal, on peut exécuter un script en écrivant « ./ » avant le nom du fichier. Le fichier « *Allclean* » est désigné pour supprimer les résultats d'exécutions, mais tout en le laissant exécutable. Un dossier « 0.org » sauvegarde les conditions aux limites :

```
./Allclean
```

Pour exécuter toute les commandes précédentes, il suffit d'écrire :

```
./Allrun
```

Les commandes précédentes sont exécutées par ordre, sauf pour la commande « paraFoam » qui n'est pas dans le script.

1.8. Conclusion

OpenFOAM est un outil open-source orienté CFD, intelligent et puissant qui fonctionne sous Linux. Cet outil gratuit est capable de résoudre plusieurs types d'écoulement, mettant en jeu de la combustion comme de la cavitation, aussi rapidement que les logiciels commerciaux. C'est pourquoi cet outil est très intéressant pour les entreprises qui veulent réduire leurs coûts de licence de logiciels CFD sans avoir recours aux sociétés commerciales. Cependant, OpenFOAM est un outil complexe à mettre en œuvre, ce qui peut décourager certains utilisateurs potentiels. Ce document se propose une bref présentation d'OpenFOAM afin de démontrer sa fiabilité sur des cas simples, et de donner au lecteur les informations importantes qui lui permettront d'utiliser OpenFOAM dans les meilleurs conditions.

CHAPITRE 2

CHAPITRE 2

ÉCOULEMENT LAMINAIRE BIDIMENSIONNEL AUTOUR D'UN CYLINDRE

2.1. Introduction

L'écoulement autour d'un cylindre est un problème classique dans la mécanique des fluides et l'aérodynamique. C'est un problème de grande importance dans la pratique car les écoulements se font autour des structures cylindriques. Le décollement de tourbillon est un phénomène important qui se produit dans l'écoulement autour de ces types de structures pour une plage de nombres de Reynolds. La fréquence du décollement des tourbillons et l'analyse des tourbillons qui induit des vibrations, ont un rôle important dans la conception de ces structures. Dans ce chapitre un écoulement autour d'un cylindre est simulé avec OpenFOAM et les résultats sont validés en les comparant aux données expérimentales.

Tout d'abord, l'aspect physique de l'écoulement autour d'un cylindre est passé en revue. Les aspects numériques de la simulation du cas ainsi que les paramètres d'exécution du cas dans OpenFOAM seront discutés. Ce chapitre finira avec la validation des résultats par la comparaison avec les données expérimentales.

L'écoulement réel d'un fluide est certainement plus compliqué que celui d'un écoulement potentiel, la différence est due principalement à l'influence des frottements et le coefficient de traîné pour le vrai écoulement n'est certainement pas zéro. Pour un écoulement incompressible visqueux, les résultats expérimentaux montrent clairement que le coefficient de traîné est une fonction du nombre de Reynolds.

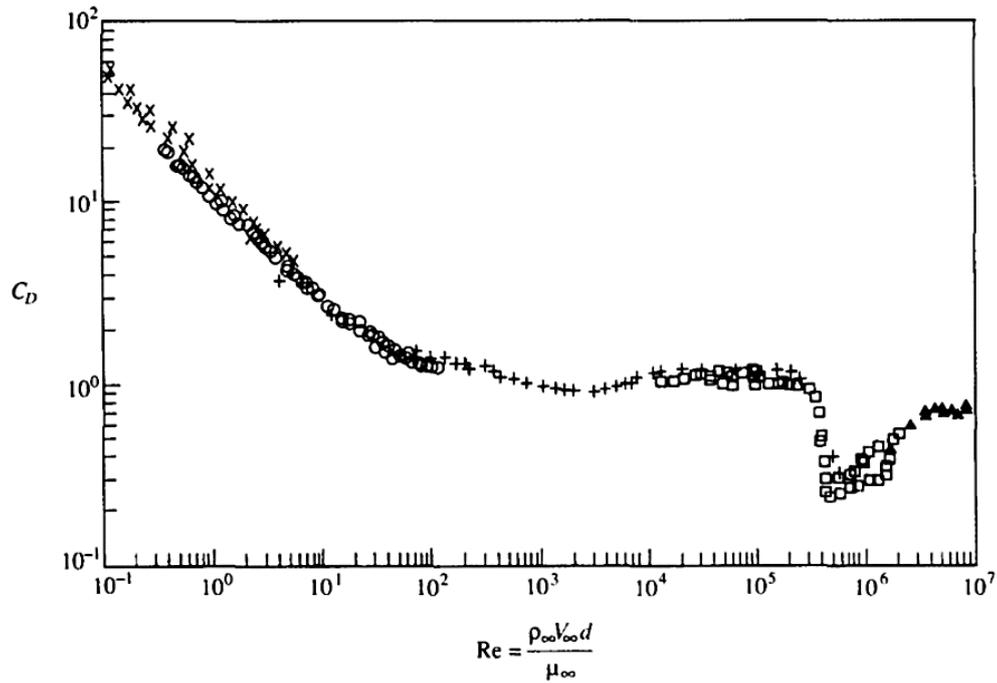
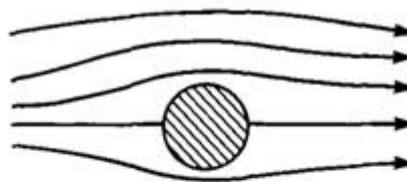


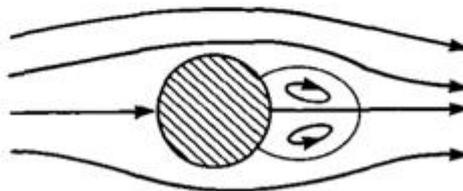
Figure 2.1 : La variation de C_x pour un cylindre circulaire en fonction de Re [10].

2.2. Physique du problème

La variation représentée sur le graphe du C_x en fonction du nombre de Reynolds est accompagné par des variations énormes sur les aspects de l'écoulement comme on le voit sur les figures 2.2.



(a)



(b)

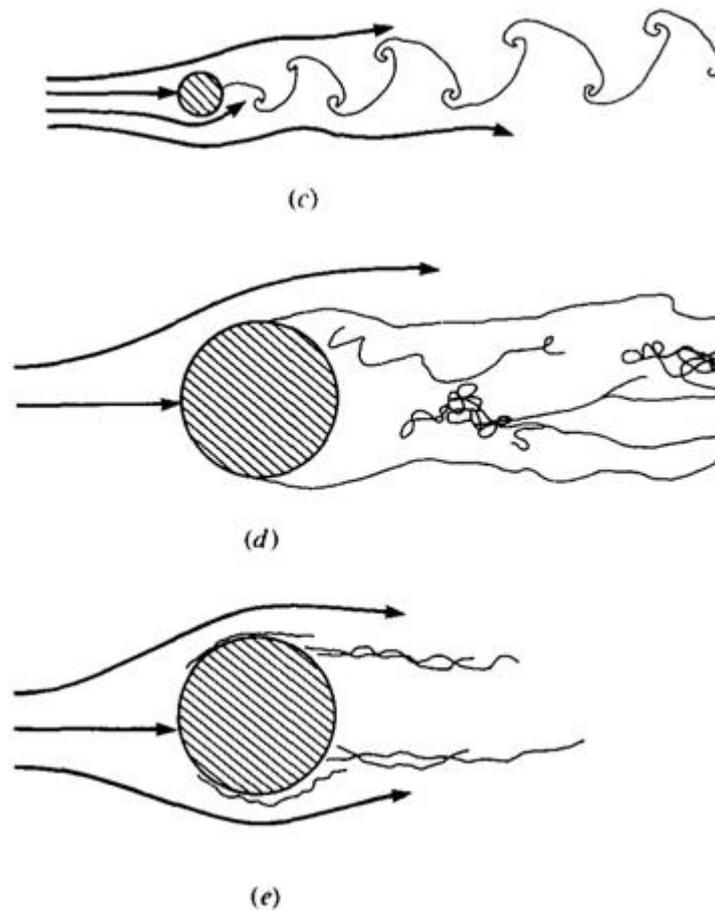


Figure 2.2 : Régimes d'écoulements visqueux autour d'un cylindre circulaire [10].

Noter que le C_x est très grand pour des valeurs extrêmement petites du Reynolds ($Re < 1$), mais diminue jusqu'à un nombre de Reynolds égal à 300 000. A partir de ce nombre, il y a une chute de C_x d'une valeur près de 1 à environ 0.3, puis un léger accroissement à environ 0.6 pour $Re = 10^7$.

- **Cas (a)** : Pour des valeurs très petites du nombre de Reynolds par exemple $0 < Re < 4$, les lignes profilées sont presque (mais pas exactement) symétriques, et l'écoulement est attaché, comme présenté dans la Figure 2.2(a). Ce régime d'écoulement visqueux s'appelle l'écoulement de Stokes ; il est caractérisé par un équilibre proche des forces de pression avec les forces de frottement agissant sur le fluide ; la vitesse d'écoulement est si basse que les effets d'inertie sont très petits. Une photographie de ce type d'écoulement est montrée sur la Figure 2.3

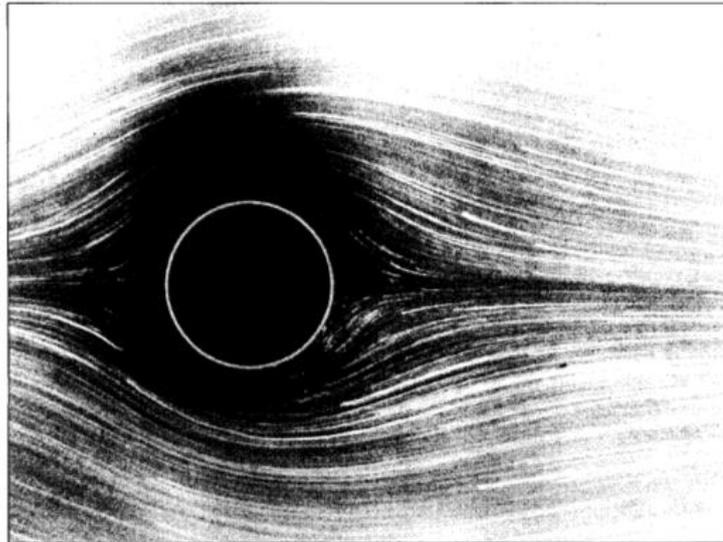


Figure 2.3 : Écoulement de l'eau autour d'un cylindre circulaire pour $Re = 1.54$ [10].

- **Cas (b)** : Pour $4 < Re < 40$, l'écoulement se sépare sur le dos du cylindre, formant deux tourbillons stables qui demeurent dans la position représentée sur la figure 2.2(b). Une photo de ce type d'écoulement est donnée sur la figure 2.4.

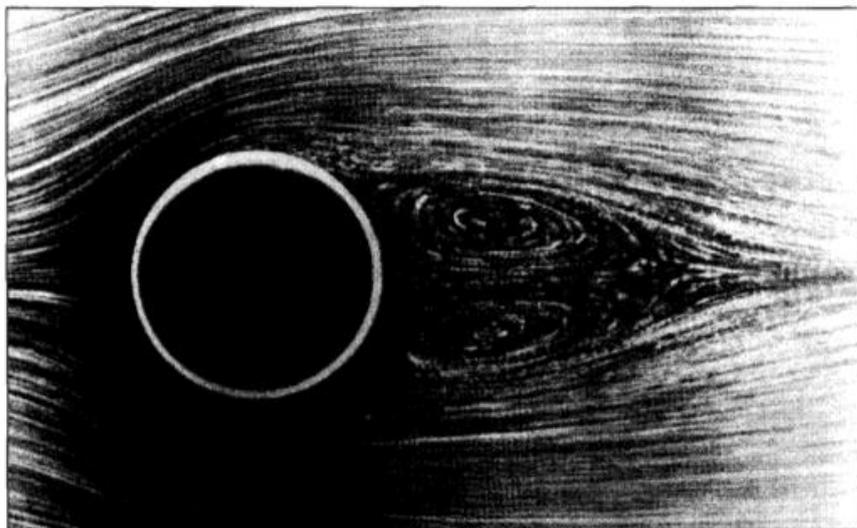


Figure 2.4 : Écoulement de l'eau autour d'un cylindre circulaire où $Re = 26$ [10].

- **Cas (c)** : Quand le nombre de Reynolds dépasse 40, l'écoulement derrière le cylindre devient instables, les tourbillons qui étaient attachés au cylindre se font d'une façon régulière et coulent en aval. Cet écoulement est montré dans la figure 2.2 (c). Une photo de ce type d'écoulement est montrée sur la figure 2.4 avec

$Re = 140$. Le modèle représenté s'appelle le vortex de Von Karman, qui porte le nom du chercheur aérodynamicien Theodore Von Karman.



Figure 2.5 : Allée de Von Karman derrière un cylindre où $Re = 140$ [10].

- **Cas (d)** : Quand le nombre de Reynolds augmente les tourbillons de Von Karman commence à se métamorphoser et forme un sillage. La couche limite laminaire sur le cylindre se sépare de la face avant, à un point environ 80° du point d'arrêt. Ceci est montré dans la figure 2.2(d). La valeur du nombre de Reynolds pour cet écoulement est de l'ordre de 10^5 . On remarque dans la figure 2.1, que le C_x a une valeur relativement constante près de l'unité pour $10^3 < Re < 3 \times 10^5$.

- **Cas (e)** : Pour $3 \times 10^5 < Re < 3 \times 10^6$, la séparation de la couche limite laminaire a lieu toujours dans la face avant du cylindre. Cependant, au-dessus de la région séparée, la transition à l'écoulement turbulent a lieu. Le flux se rattache alors sur la face arrière du cylindre, mais se sépare encore à environ 120° mesuré à partir du point d'arrêt. Cet écoulement est esquissé dans la figure 2.2(e). Cette transition à l'écoulement turbulent réduit la traînée de pression sur le cylindre et est la cause de la baisse brusque du C_x lorsque $Re = 3 \times 10^5$.

- **Cas (f)** : Pour $Re < 3 \times 10^6$, la couche limite transite directement à l'écoulement turbulent à un certain point sur la face avant, et demeure totalement attachée à la surface jusqu'à ce qu'elle se sépare à un emplacement angulaire légèrement moins de 120° depuis le point d'arrêt. Pour ce régime d'écoulement, le

C_x augmente légèrement avec l'augmentation du Re parce que les points de séparation, sur la surface arrière, commencent à se rapprocher vers le haut et le bas du cylindre, produisant un plus gros sillage et par conséquent une plus grande traînée de pression.

2.3. Equations gouvernant l'écoulement

Dans ce chapitre, il n'y a aucune solution analytique facile pour décrire le comportement du fluide (sauf l'écoulement potentiel, qui est un cas idéal). Cependant, il est nécessaire d'avoir dans l'esprit les équations principales et les nombres sans dimensions qui concernent le problème :

L'équation de continuité :

$$\nabla \cdot U = 0 \quad (2.1)$$

L'équation de conservation de quantité de mouvement :

$$\frac{\partial U}{\partial t} + U \cdot \nabla U = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 U \quad (2.2)$$

Le nombre de Reynolds :

$$Re = \frac{D|U|}{\nu} \quad (2.3)$$

Le nombre de Strouhal, qui décrit les mécanismes d'oscillation d'écoulement :

$$St = \frac{fD}{|U|} \quad (2.4)$$

Le coefficient de traînée, qui est un nombre sans dimension dont on remarque l'influence de la force aérodynamique sur le cylindre :

$$C_D = \frac{F_D}{\frac{1}{2}\rho|U|^2 S} \quad (2.5)$$

La fréquence des oscillations qui décrit la répétition dans un intervalle de temps appelé la période. Elle est définie par :

$$f = \frac{1}{T} \quad (2.6)$$

2.4. Résolution numérique

Cas d'un cylindre ou on a $R = 1\text{ m}$ submergé dans un jet avec une vitesse de $U = 1\text{ m/s}$ et une pression ambiante. Le problème se présente comme montré dans la figure 2.6.

Hypothèses :

- Écoulement incompressible.
- Écoulement laminaire.
- Écoulement newtonien.
- Écoulement bidimensionnel.
- Effets de gravité négligeables.

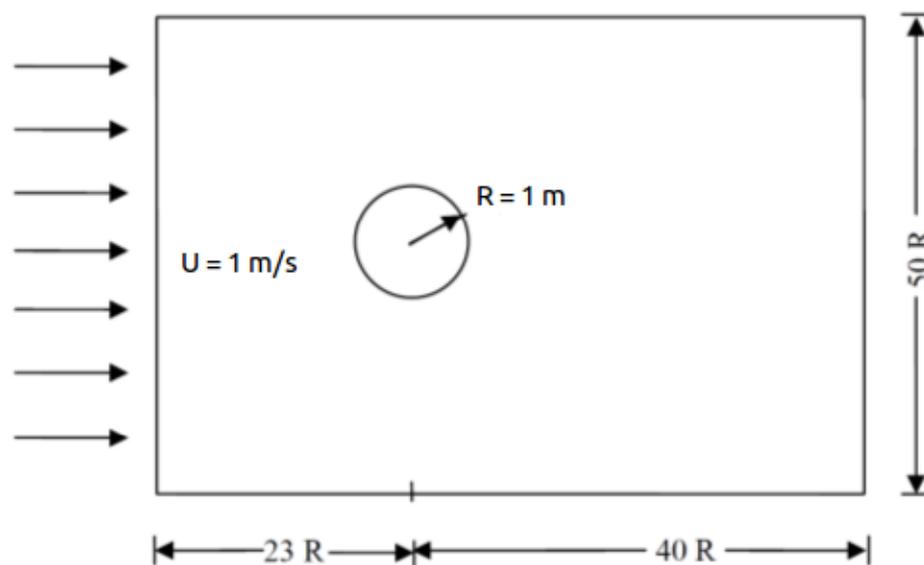


Figure 2.6 : Schéma du problème.

2.4.1. Prétraitement

Les codes suivants contiennent des informations pour simuler avec $Re = 100$ par le solveur `pisoFoam` et avec un maillage généré par `blockMesh`. Sa structure

des fichiers et des sous-dossiers est très similaire à celle utilisée au chapitre précédent :

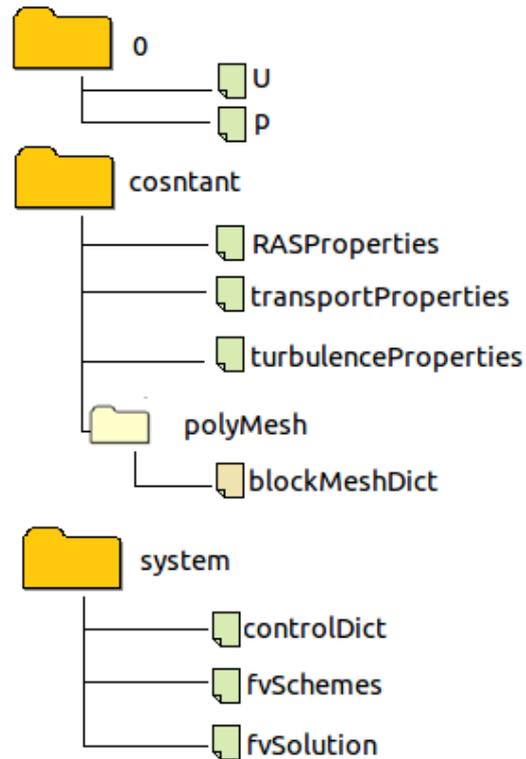


Figure 2.7 : Structure du cas étudié.

Le maillage pour cette étude ne va pas être uniforme. Quelques régions du domaine doivent contenir une densité plus élevée de mailles que d'autres (principalement au niveau de la paroi du cylindre). Dans ce cas il est nécessaire de diviser le domaine en différents blocs comme dans la Figure 2.8. Ceci nous permet de choisir les zones à raffiner.

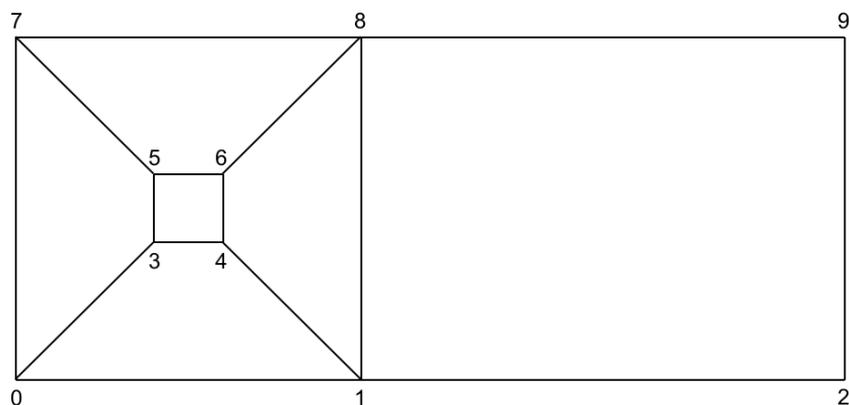


Figure 2.8 : Représentation des zones de maillages.

Pour générer un tel maillage, on commence par écrire les points par ordre de 0 à 9. Ensuite, par symétrie par rapport au plan (x, y), 10 autres points seront ajoutés. Les liaisons entre les points 3, 4, 5 et 6 sont des droites par défaut, pour dessiner un cercle, il faut ajouter des arcs de liaison.

2.4.2. Préparation du fichier blockMeshDict

Dans le dossier de cas, le fichier constant/plyMesh/blockMeshDict doit contenir les instructions suivantes :

```

/*----- C++ -----*/
|=====|
| \ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \ / Operation | Version: 2.3.0 |
| \ / And | Web: www.OpenFOAM.org |
| \ / Manipulation |
|-----*/
FoamFile
{
  version 2.0;
  format ascii; // Format du fichier
  class dictionary; // Type du fichier
  object blockMeshDict; // Nom du fichier
}
// ***** // // Ligne d'ouverture
convertToMeters 1;
vertices // Les points de la géométrie
(
  (-23 -25 0) //0
  ( 23 -25 0) //1
  (40 -25 0) //2
  (-0.7071 -0.7071 0) //3
  ( 0.7071 -0.7071 0) //4
  (-0.7071 0.7071 0) //5
  ( 0.7071 0.7071 0) //6
  (-23 25 0) //7
  ( 23 25 0) //8
  (40 25 0) //9
  (-23 -25 1) //10
  ( 23 -25 1) //11
  (40 -25 1) //12
  (-0.7071 -0.7071 1) //13
  ( 0.7071 -0.7071 1) //14
  (-0.7071 0.7071 1) //15
  ( 0.7071 0.7071 1) //16
  (-23 25 1) //17
  ( 23 25 1) //18
  (40 25 1) //19
);

```

```

blocks                                     // Les blocs de la géométrie
(
    hex (0 3 5 7   10 13 15 17) (200 60 1) simpleGrading (0.20 1 1) // Le bloc 0
    hex (1 4 3 0   11 14 13 10) (200 60 1) simpleGrading (0.20 1 1) //      1
    hex (4 1 8 6   14 11 18 16) (200 100 1) simpleGrading (5 1 1) //      2
    hex (6 8 7 5   16 18 17 15) (200 60 1) simpleGrading (5 1 1) //      3
    hex (1 2 9 8   11 12 19 18) (100 100 1) simpleGrading (1 1 1) //      4
);
edges                                     // Les courbes de liaison :
(
    arc3 5 (-1 0 0) // Arc de 3 a 5, passant par le point (-1 0 0)
    arc13 15 (-1 0 1)
    arc 4 6 (1 0 0)
    arc14 16 (1 0 1)
    arc 5 6 (0 1 0)
    arc15 16 (0 1 1)
    arc3 4 (0 -1 0)
    arc13 14 (0 -1 1)
);
boundary                                  // Les surfaces de la géométrie
(
inlet                                     // La face amont de l'écoulement
{
    type patch;
    faces
    (
        (17 7 0 10) // Une face est définie par 4 points
    );
}
outlet                                    // La face aval de l'écoulement
{
    type patch;
    faces
    (
        (9 19 12 2)
    );
}
top                                        // La face supérieur de la géométrie
{
    type wall;
    faces
    (
        (8 7 17 18)
        (9 8 18 19)
    );
}
bottom                                    // La face inférieur de la géométrie
{
    type wall;
    faces
    (
        (0 1 11 10)
        (1 2 12 11)
    );
}
}

```

```

cylinder                                // Les faces du cylindre
{
    type wall;
    faces
    (
        (5 6 16 15)
        (16 6 4 14)
        (4 3 13 14)
        (3 5 15 13)
    );
}
sides                                    // Les faces avant et arrières de la géométrie
{
    type empty;
    faces
    (
        (7 8 6 5)
        (7 5 3 0)
        (3 4 1 0)
        (8 1 4 6)
        (8 9 2 1)
        (18 17 15 16)
        (15 17 10 13)
        (14 13 10 11)
        (18 16 14 11)
        (19 18 11 12)
    );
}
);
mergePatchPairs
(
);
// ***** //
// Ligne de fermeture

```

Le maillage généré par blockMesh est toujours structuré et les mailles sont de forme hexagonale. Pour exécuter blockMesh, il suffit d'accéder au dossier dans le Terminal et taper la commande « blockMesh ».

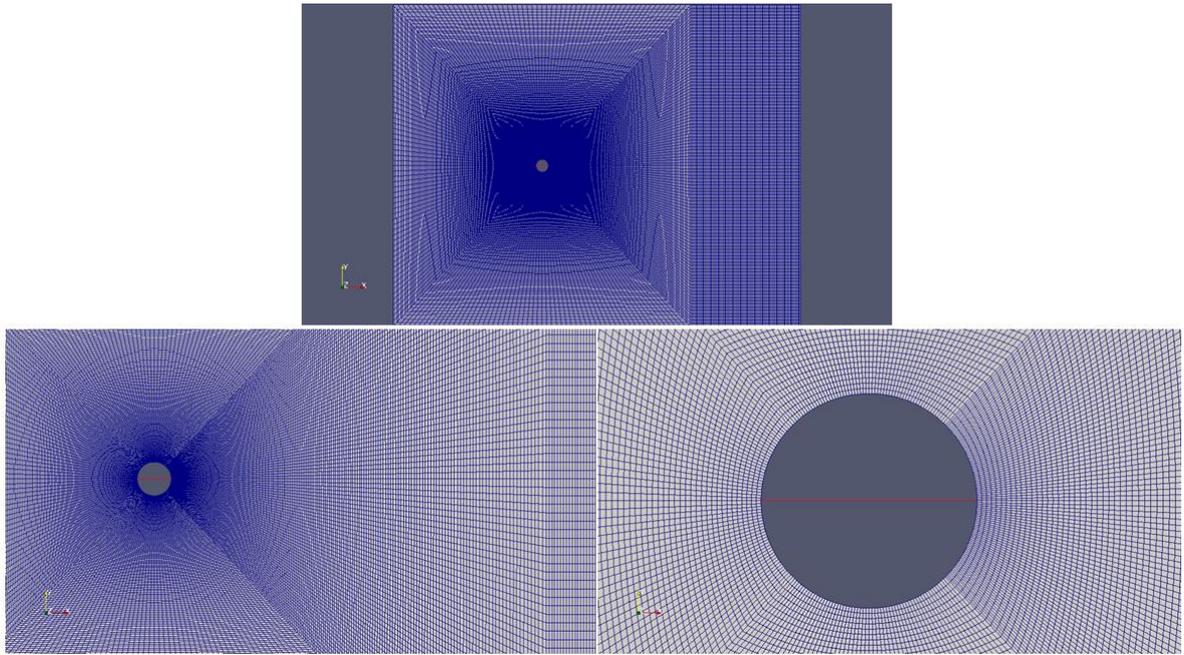


Figure 2.9 : Visualisation du maillage

2.4.3. Le traitement

La partie de procession, ou de traitement, consiste à faire le choix du solveur ainsi que les conditions aux limites pour la simulation. Cette étape n'est pas indépendante des autres, à vrai dire, le choix du maillage dépend du solveur. Et ainsi, les régions de raffinement sont choisies.

2.4.3.1. Choix du solveur pisoFoam

Le choix du solveur dépend principalement du type du problème et de ces hypothèses. PisoFoam est un solveur pour les écoulements incompressibles, laminaires et turbulents. Il est basé sur l'algorithme de PISO (Pressure Implicit with Splitting of Operators), proposé par l'ISSA en 1995. C'est un algorithme de couplage de pression-vitesse avec une étape de prédiction et deux étapes de corrections. Dans l'étape de prédiction, les équations de quantité de mouvement sont résolues pour un champ intermédiaire de pression. Le champ estimé de vitesse à ce stade ne remplit pas la condition de continuité. Alors pendant les deux étapes de corrections, les champs de vitesse et de pression sont corrigés d'une manière à vérifier tous les deux les équations de continuité et de quantité de mouvement.

2.4.3.2. Les conditions aux limites

Le dossier 0 contient les paramètres physiques qui gouvernent les équations d'écoulement. Dans ce cas, la vitesse U et la pression p . Tandis que le fichier constant/transportProperties contient les constantes qui caractérisent le fluide étudié. Six faces élémentaires sont définies dans ce problème : inlet, outlet, top, bottom, cylinder. Les faces supérieur et inférieur sont définies d'une façon à être assez loin du cylindre, elles sont définies comme des murs de glissade pour ne pas affecter l'écoulement. Puisque c'est un cas 2D, les murs latéraux sont supposés vides. Les conditions aux limites utilisées sont récapitulées dans le tableau 2.1 :

Tableau 2.1 : Conditions aux limites.

Paramètres	Faces	Conditions aux limites	Description
U	inlet	fixedValue, uniform (1 0 0)	Vitesse uniforme 1m/s.
	outlet	zeroGradient	Gradient nulle dans la direction de la normale.
	cylinder	fixedValue, uniform (0 0 0)	Vitesse fixe (nulle).
	top	slip	Condition de glissement de la paroi.
	bottom	slip	//
	sides	empty	Faces vides
p	inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	outlet	fixedValue, uniform 0	Pression fixe (nulle).
	cylinder	zeroGradient	Gradient nulle dans la direction de la normale.
	Top	zeroGradient	//
	Bottom	zeroGradient	//
	Sides	empty	Faces vides

2.4.4. Post-traitement

Dans la présente partie, les résultats de la simulation sont évalués et comparés aux résultats de référence. L'écoulement initial a besoin d'un certain temps pour se développer autour du cylindre, après un moment, le début du décollement de tourbillon se produit et on peut voir le développement de l'allée de Von Karman (figure 2.10). Le changement des coefficients de portance et de traîné au cours du temps peut être vu sur les figures 2.11 et 2.12. Le coefficient de traîné commence à osciller pour une valeur moyenne de 1,386 et l'amplitude du coefficient de portance est 0,358. Elles sont dans une concordance très bonne avec les valeurs référer 1,39 et 0,35.

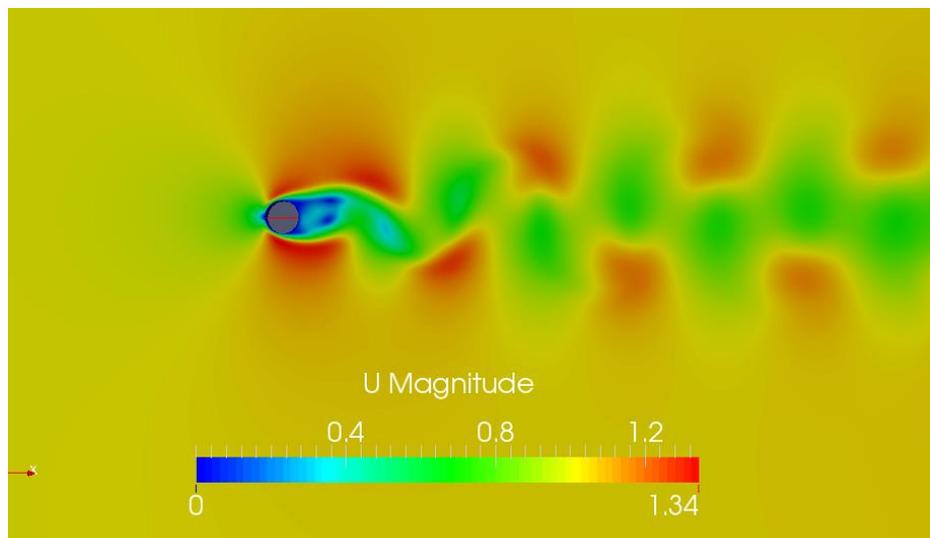


Figure 2.10 : Visualisation du champ de vitesse et l'allée de Von Karaman.

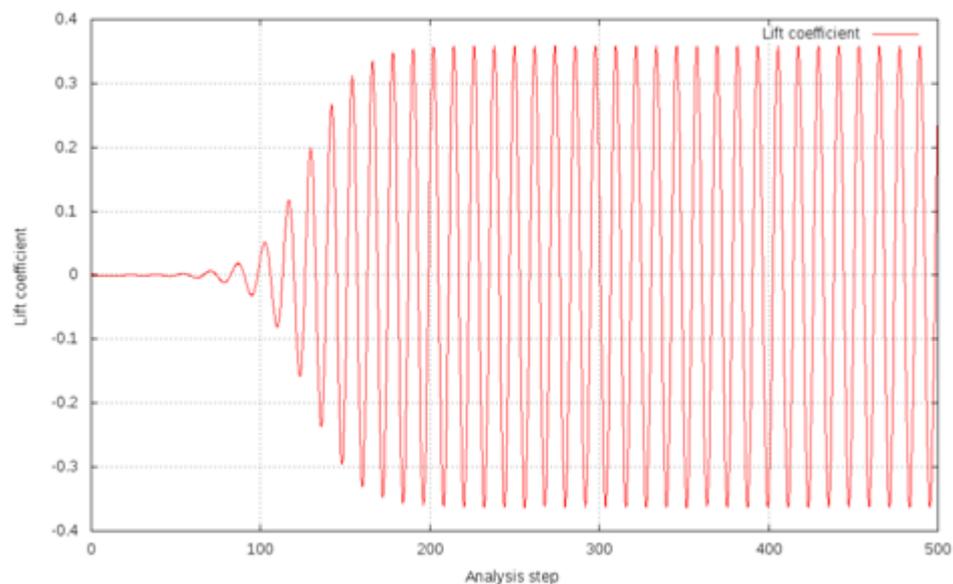


Figure 2.11 : Évolution du coefficient de portance.

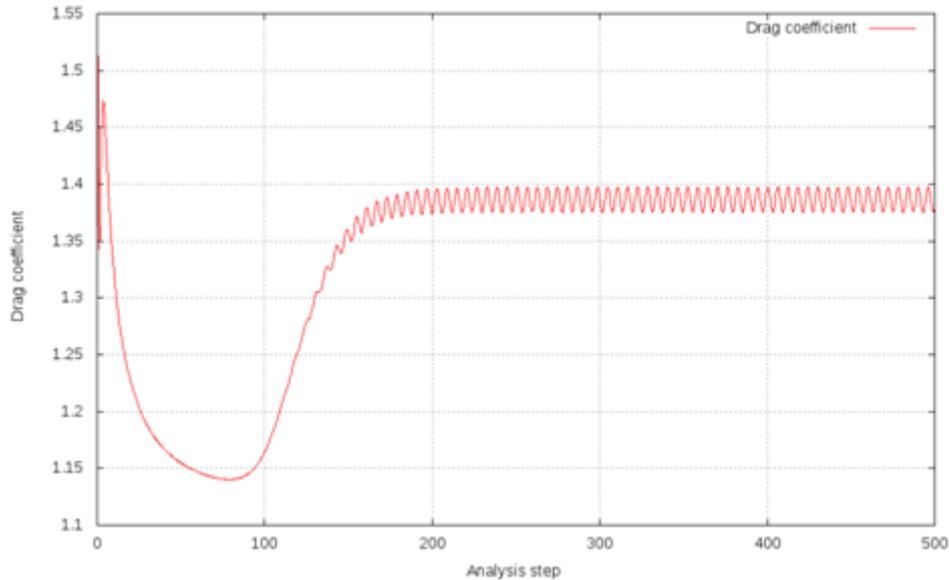


Figure 2.12 : Évolution du coefficient de traîné.

La prochaine quantité à valider est le nombre de Strouhal, qui décrit la fréquence du décollement des tourbillons. Depuis le fichier des résultats des coefficients calculés, on cherche la durée ou le résultat se répète, cet intervalle de temps est appelé la période T .

D'après le graphe de C_z , la fréquence des oscillations est :

$$f = \frac{1}{12} = 0,0833 \text{ s}^{-1}$$

Le nombre correspondant de Strouhal peut être calculé à partir de l'équation 2.4 :

$$St = \frac{fD}{U} = \frac{0.833 \times 2}{1} = 0.166$$

Les valeurs trouvées sont conformes avec la valeur 0,17 comme le montre la figure 2.1.

2.4.5. Lignes de courants à $Re = 2, 20$ et 100

D'autres simulations ont été faites pour différents nombres de Reynolds afin de visualiser les changements de phase dans ce type d'écoulement.

Pour $Re = 2$ les lignes de courant sont attachées à la paroi du cylindre. Pour obtenir ces résultats il suffit de changer la vitesse de l'écoulement précédent :

$$U = 0,02 \text{ m/s.}$$

```
inlet
{
  typefixedValue;
  value      uniform (0.02 0 0);
}
```

Pour $Re = 20$ les lignes de courants forment deux tourbillons jumeaux derrière le cylindre. Tout comme le cas précédent, la seule la vitesse d'entrer d'écoulement. a changé Pour $Re = 20$, la vitesse sera $U = 0,2 \text{ m/s.}$

```
inlet
{
  typefixedValue;
  value      uniform (0.2 0 0);
}
```

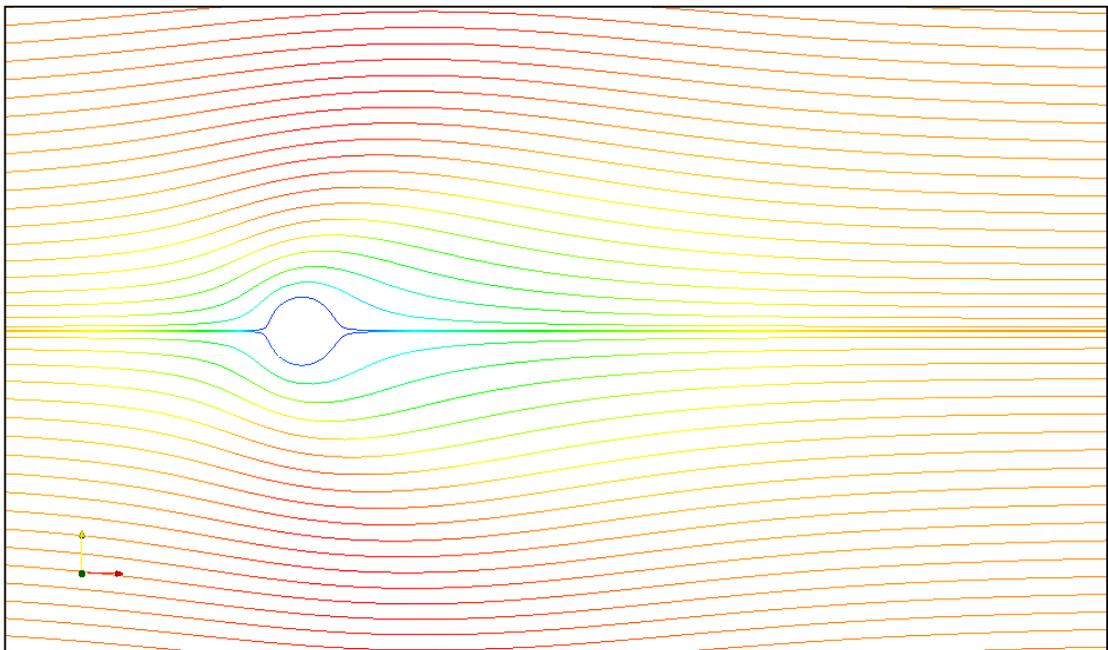


Figure 2.13 : Visualisation des lignes de courants à $Re = 2$.

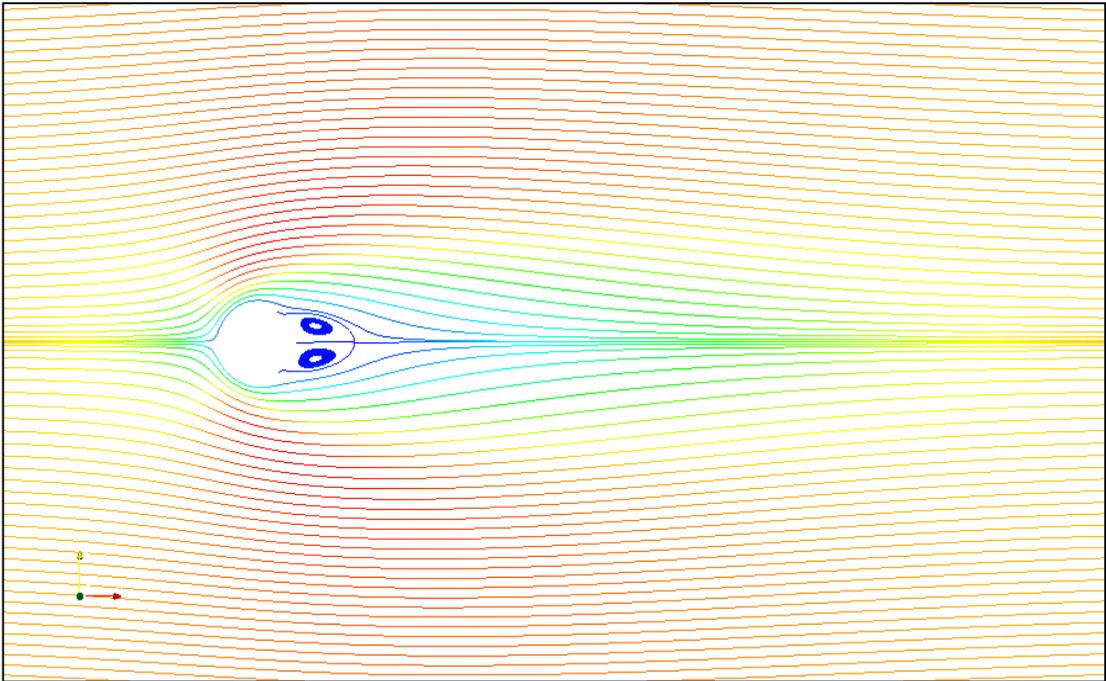


Figure 2.14 : Visualisation des lignes de courants à $Re = 20$.

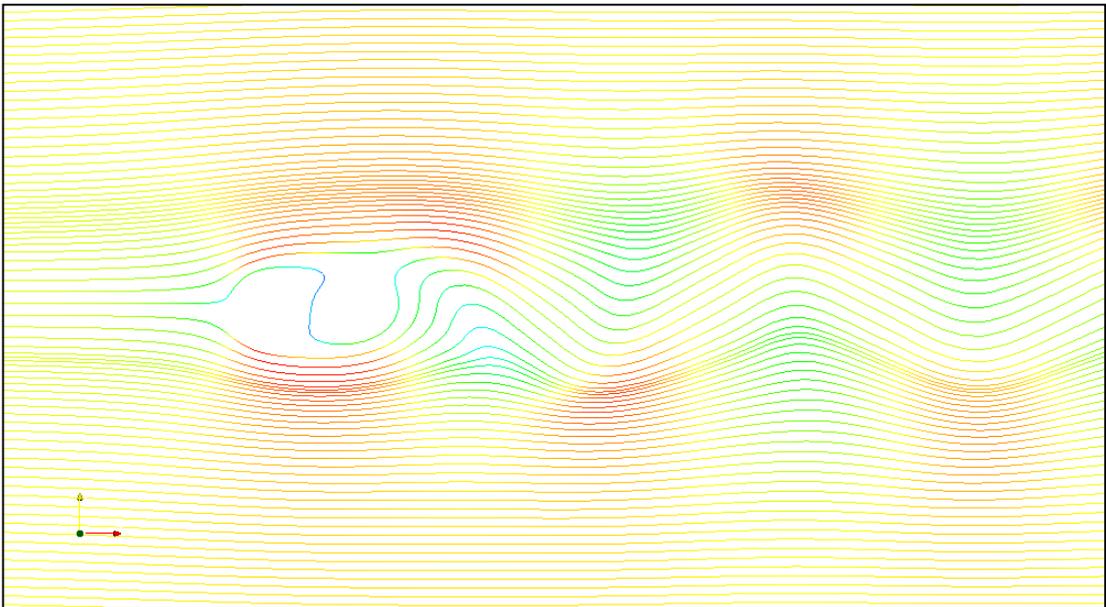


Figure 2.15 : Visualisation des lignes de courants à $Re = 100$.

2.5. Conclusion

L'étude de l'écoulement autour d'un cylindre est fondamentale dans l'aérodynamique. L'étude de ce problème nous permet à comprendre des phénomènes curieux qui se passent derrière ce type de structure.

Dans ce chapitre, une analyse d'un écoulement laminaire autour d'un cylindre est faite à l'aide d'OpenFOAM puis comparées avec les résultats expérimentaux. Le calcul des coefficients de portance et de traîné nous ont permis d'évaluer le nombre de Strouhal.

CHAPITRE 3

CHAPITRE 3

ÉCOULEMENT AUTOUR D'UN PROFIL D'AILE

3.1. Introduction

L'aile portante d'envergure finie est un élément très important dans la technologie de nos jours, on l'utilise dans le domaine aérospatial (aile d'avion, pales d'hélicoptère, ailes d'une navette spatiale), en construction navale (pales d'hélices de bateaux, gouvernails) ou dans la génération d'énergie (aérogénérateur : pale des éoliennes). Pour produire une portance, les ailes ont une forme spéciale et elles sont profilées, ce qui contribue à la diminution du frottement avec le fluide, c'est-à-dire à la réduction de la traînée. C'est aussi l'élément qui crée la force de sustentation, c'est la force verticale qui permet de contrer le poids de l'avion.

Cette force est due à la différence de pression qui s'établit entre les deux côtés de l'aile (Une surpression sous l'aile, à l'intrados, et surtout une forte dépression au-dessus, à l'extrados). C'est un domaine de recherche très vaste, beaucoup d'aérodynamiciens sont occupés à faire des calculs pratiques et numériques afin de trouver le profil idéal qui leur correspond car c'est selon les caractéristiques aérodynamiques du profil qu'on décide de son domaine d'application.

Dans le présent chapitre, on a essayé de faire deux simulations numériques d'écoulements incompressibles bidimensionnels pour deux profils différents, Le premier concerne un profil de Wortmann FXS02-196 ou a $Re=122205$, les résultats trouvés seront comparés avec les résultats expérimentaux de la soufflerie, le deuxième est le profil NACA4412 ou on a $Re=3000000$ dans ce cas les résultats seront comparés avec celle de la littérature. Le but de ce chapitre est donc double :

- *Utiliser deux approches de turbulences sur OpenFOAM pour résoudre les problèmes des bas et des hauts nombres de Reynolds.*
- *Comparer les résultats pratiques de la soufflerie réelle avec la soufflerie numérique d'OpenFOAM.*

3.2. Description physique

Dans le ce chapitre, il n'existe aucune solution analytique qui d'écrit l'écoulement réel autour du profil, sauf quelque méthodes théoriques avec des hypothèses simplificatrice. Le calcul des paramètres aérodynamiques dépend principalement des études de soufflerie. Malgré cela, il faut prendre en considération les équations de la mécanique de fluide qui gouvernent cet écoulement.

L'équation de continuité,

$$\nabla \cdot U = 0 \quad (3.1)$$

L'équation de quantité de mouvement,

$$\frac{\partial U}{\partial t} + U \cdot \nabla U = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 U \quad (3.2)$$

Le nombre de Reynolds,

$$Re = \frac{c|U|}{\nu} \quad (3.3)$$

Le coefficient de portance,

$$C_z = \frac{F_z}{\frac{1}{2}\rho|U|^2S} \quad (3.4)$$

Le coefficient de trainé,

$$C_x = \frac{F_x}{\frac{1}{2}\rho|U|^2S} \quad (3.5)$$

Le coefficient de pression,

$$C_p = \frac{p-p_\infty}{\frac{1}{2}\rho_\infty|U_\infty|^2} \quad (3.6)$$

3.3. Le profil de WortmannFXS02-196

C'est un profil conçu pour les faibles nombres de Reynolds, il équipe des planeurs ou des avions radio commandés des basses vitesses comme l'« Alexander Schleicher ASK 21 » et le « SZD-51 Junior ». Pour valider nos résultats, nous avons profité d'une période de stage chez l'ERMA, à l'URD pour faire un test sur une soufflerie subsonique. Cette soufflerie est équipée d'une balance pour la mesure des forces de traînée et de portance. Les résultats expérimentaux obtenus seront comparés avec ceux de la simulation. Dans ce chapitre on va analyser les caractéristiques aérodynamiques d'un profil d'aile et comparer les résultats obtenus dans les deux souffleries : la soufflerie réelle et la soufflerie numérique d'OpenFOAM. Le problème s'agit donc d'une aile de type FXS02-196 d'envergure supposée infinie et d'une corde C qui évolue avec une vitesse U dans les conditions atmosphériques. Comme le milieu est supposé incompressible ($M \ll 0,3$) la masse volumique de l'air est considéré comme étant égale à l'unité $\rho = 1,225 \text{ kg/m}^3$ et la viscosité cinématique $\mu = 1,79 \times 10^{-5} \text{ kg/m.s}$

Pour qu'il y ait une ressemblance dans les deux expériences (numériques et expérimentales), d'après l'équation (3.3), le nombre de Reynolds sera choisi fixe :

$$Re = \frac{\rho C |U|}{\mu} = 122203$$

Le profil de Wortmann a été choisi pour ses bonnes performances aérodynamiques pour les basses vitesses, c'est un profil très intéressant car il peut voler sur une large plage d'incidence.

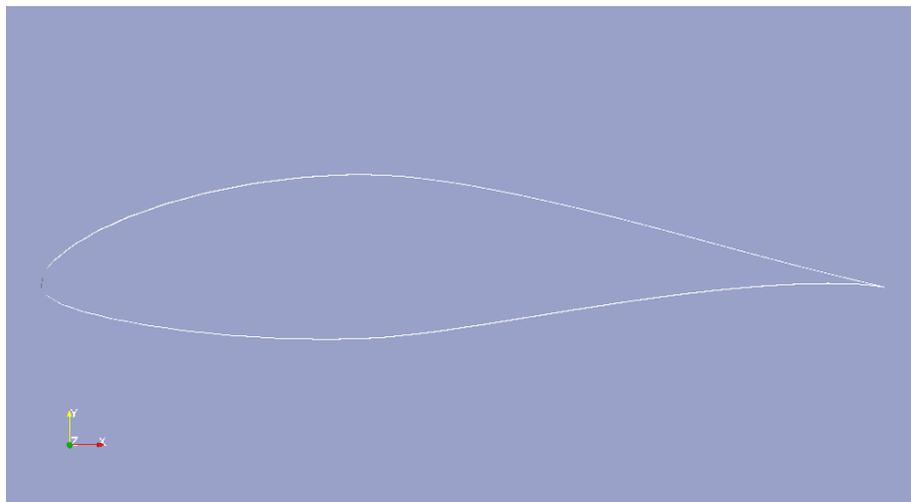


Figure 3.1 : Profil de Wortmann FXS02-196.

Pour les faibles vitesses, et avec une rugosité de surface donnée, C_x et C_z dépendent uniquement de l'angle d'attaque et du nombre de Reynolds.

$$C_z = f(Re, \alpha)$$

$$C_x = f(Re, \alpha)$$

Où α est l'angle entre la corde du profil et la direction de la vitesse d'écoulement. A des faibles incidences, le gradient de pression dans l'extrados n'est pas suffisamment grand pour détacher la couche limite, les lignes de courants sont attachées et la portance varie presque linéairement avec l'angle d'attaque. Quand α augmente, le gradient de pression croît, et à environ 20° l'écoulement se détache complètement du profil. On appelle le premier angle de détachement d'écoulement l'angle de décrochage.

3.3.1. Partie expérimentale

Pendant une période d'un mois de stage chez l'Entreprise de Rénovation des Matériels Aéronautique (ERMA), à l'Unité de Recherche et Développement (URD), nous avons eu l'occasion de faire un test sur une soufflerie pédagogique. Dans cette partie, on a essayé de faire une étude pratique sur le profil de Wortmann FXS02-196. Tout d'abord le profil est créé avec du Polystyrène sur une machine de découpe numérique destiné au découpage par fil chaud (Figure 3.2). Le profil est ensuite renforcé par un tissu de verre, de la résine et un durcisseur. L'encastrement est fait avec une plaque d'aluminium et une tige filetée de 4mm percée au premier quart de la corde.



Figure 3.2 : Machine de découpe numérique par fil chaud.

Cette petite aile a une corde de 11mm et une envergure de 290mm. Ces dimensions conviennent très bien avec les dimensions de la veine d'essai, l'envergure est choisie de telle façon que sa longueur ne soit ni trop petite ni trop grande pour quelle n'interagit pas avec la paroi. La corde et l'épaisseur maximale sont aussi choisies en tenant compte de la largeur de la veine d'essai.

3.3.1.1. Description de la soufflerie

C'est une soufflerie subsonique à circuit ouvert, composée essentiellement de 3 parties :

- a) Section convergente (ou collecteur) : C'est l'entrée de la soufflerie où se trouve une section en nid d'abeilles dont le rôle est d'uniformiser l'écoulement.
- b) La veine d'essai : C'est là où on met l'obstacle ou l'objet à étudier.
- c) Section divergente (ou diffuseur) : c'est la partie de refoulement de l'écoulement par l'hélice du moteur.

❖ Caractéristiques

C'est une soufflerie de 2,98m de longueur, de 1,83m de hauteur et de 0,8m de largeur. Sa veine d'essai est de forme octogonale de 300mm × 450mm. Elle est équipée d'un moteur de 1,5 KW avec un variateur de vitesse qui peut générer un écoulement d'air avec une vitesse maximale égale a 26m/s. La vitesse peut être lue directement sur le tube manométrique incliné.



Figure 3.3 : Indicateur de vitesse.

Elle est équipée aussi d'une balance aérodynamique, qui nous donne une lecture directe de la force de portance et de trainée, et des tubes de Pitot pour le calcul de pression.

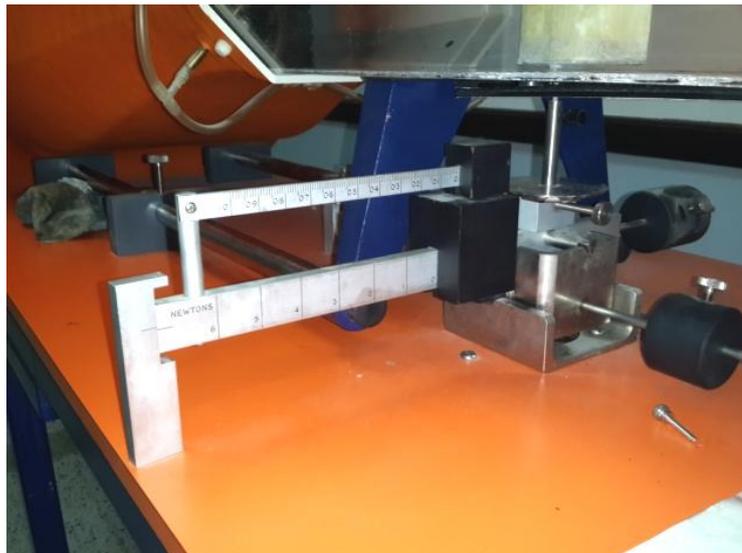


Figure 3.4 : Balance pour le calcul des forces de portance et de trainée.



Figure 3.5 : Instrument de mesure des pressions.

Le liquide utilisé dans les indicateurs est un mélange de paraffine et de colorant avec une densité de $0,784 \text{ kg/m}^3$ à $18,3^\circ\text{C}$.

❖ Fiche technique

Tableau 3.1 : Caractéristique de la soufflerie.

Longueur	2,98 m
Hauteur	1,83 m
Largeur	0,80 m
Veine d'essai	300 mm oct. × 450 mm long
Balace max – Portance	7,0 N
Balace max – Traînée	2,5 N
Sensibilité de la balance	0,01 N
Alimentation électrique	240 V / 50 ou 60 Hz
Puissance moteur	1,5 Kw

❖ Domaines d'utilisation

C'est une soufflerie consacré à l'étude des écoulements dans les petits projets de recherche comme :

- ✓ Etude de comportement d'un écoulement autour d'un corps : cylindre, sphère, hémisphère, concave ou convexe, disque circulaire, ...etc.
- ✓ Evaluation du champ de pression autour d'un profil.
- ✓ Détermination des coefficients de portance et de traînée d'un corps donné.
- ✓ Etude aérodynamique d'un modèle réduit.



Figure 3.6 : Soufflerie subsonique Armfield C2.

3.3.1.2. Description de l'expérience

Notre étude a été faite pour un nombre de Reynolds bien précis, en tenant compte de la géométrie du profil, et des caractéristiques du fluide

On a choisit pour.

$$\rho = 1,225 \text{ kg/m}^3$$

$$C = 0,11 \text{ m}$$

$$\mu = 1,79 \times 10^{-5} \text{ kg/m.s}$$

$$U = 16,22 \text{ m/s}$$

Cette vitesse est imposée à partir d'une boîte de commande des fréquences, pour notre cas, la vitesse $16,22 \text{ m/s}$ est équivalente a $34,8 \text{ Hz}$.

L'équation de Reynolds (3.3) permet d'écrire :

$$Re = \frac{\rho C |U|}{\mu} = \frac{1,225 \times 0,11 \times 16,22}{1,79 \times 10^{-5}} = 122103,07$$

3.3.1.3. Résultats obtenus

Puisqu'il s'agit d'une balance mécanique, nous somme limités par la force de portance qui doit être supérieur à zéro pour qu'on puisse l'évaluer.

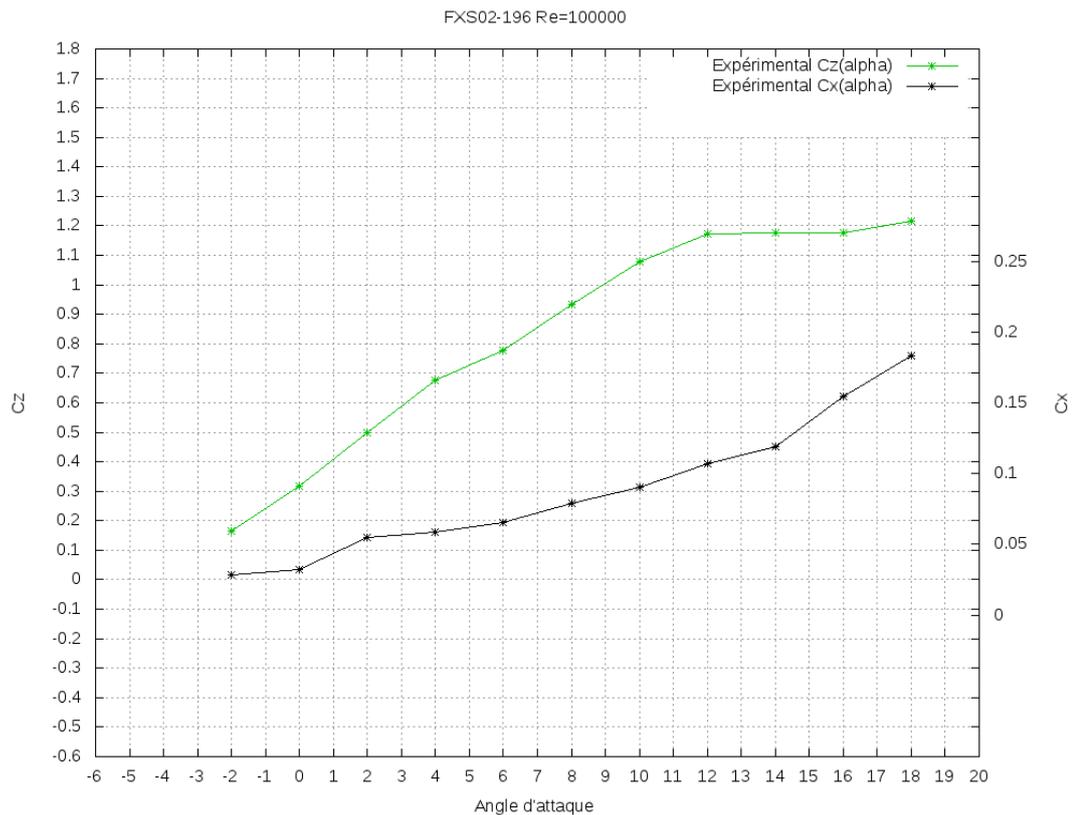
On rappelle que les coefficients de portance et de traînée peuvent être respectivement calculés grâce aux les équations 3.4 et 3.5.

$$C_z = \frac{F_z}{\frac{1}{2}\rho|U|^2S} \quad (3.4)$$

$$C_x = \frac{F_x}{\frac{1}{2}\rho|U|^2S} \quad (3.5)$$

Tableau 3.2 : Résultat expérimental.

Alpha°	Force de traîné (N)	Force de portance (N)	Cx	Cz
-2	0,12	0,69	0,02859689	0,16443214
0	0,135	1,33	0,03217151	0,3169489
2	0,23	2,09	0,05481071	0,49806256
4	0,245	2,83	0,05838532	0,67441007
6	0,275	3,26	0,06553455	0,77688228
8	0,33	3,92	0,07864146	0,93416519
10	0,38	4,53	0,09055683	1,07953273
12	0,45	4,92	0,10723835	1,17247264
14	0,5	4,93	0,11915372	1,17485571
16	0,65	4,93	0,15489984	1,17485571
18	0,77	5,1	0,18349673	1,21536798

Figure 3.7 : C_z Et C_x en fonction de l'angle d'attaque α .

Les courbes obtenus montrent bien l'évolution avec l'angle d'attaque, C_x augmente sous forme d'une parabole avec l'angle d'attaque et le C_z croît presque linéairement jusqu'à une valeur max ou il se stabilise ensuite il continue à croître légèrement.

3.3.2. Partie numérique

Chaque simulation numérique passe par trois étapes principales, le prétraitement, le traitement et le post-traitement. On a décrit dans le chapitre 1, OpenFOAM a un ensemble de solution pour chacune de ces étapes. Le présent chapitre présente une solution simple et efficace pour ce genre de problème.

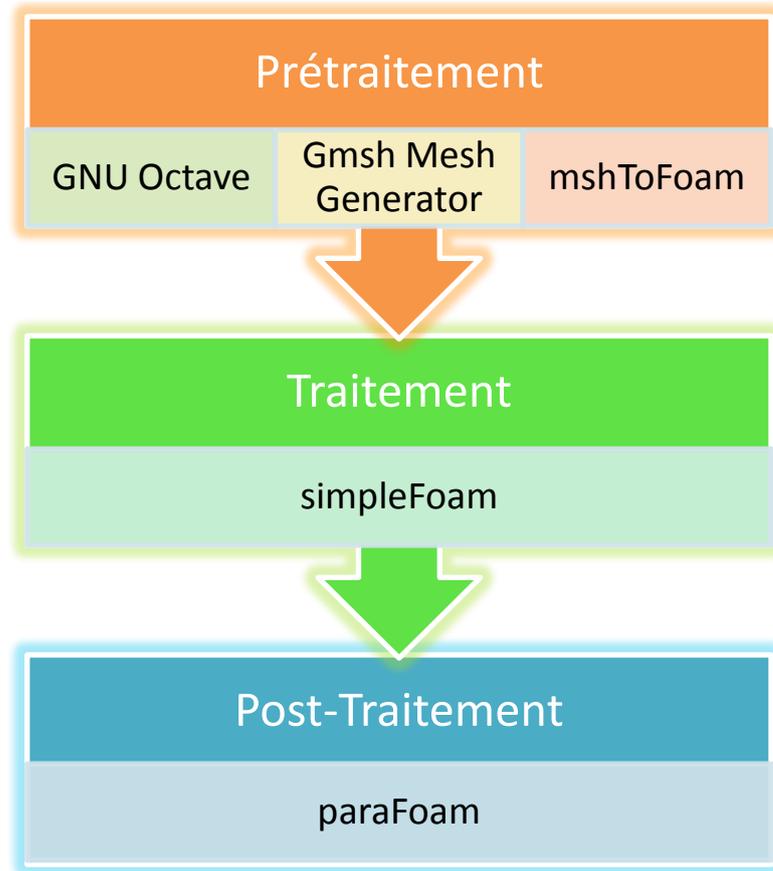


Figure 3.8 : Procédures de la simulation.

3.3.2.1. Présentation

Dans ce qui suit, on traite le cas d'un écoulement incompressible ($M \ll 0,3$), bidimensionnel (envergure infini) autour d'un profil d'aile. Le profil évolue dans des faibles vitesses et le nombre de Reynolds est supposée égale à 122103.

Hypothèses :

- Écoulement incompressible.
- Couche limite laminaire.
- Écoulement newtonien.
- Écoulement bidimensionnel.
- Effets de gravité négligeables.

On a :

$$\rho = 1,225 \text{ kg/m}^3$$

$$\vartheta = 1,46 \times 10^{-5} \text{ kg/m.s}$$

La corde du profil est égale à l'unité $C = 1 \text{ m}$ et la vitesse d'écoulement est

$$U = \frac{Re \times \vartheta}{C} = \frac{122103 \times 1,46 \times 10^{-5}}{1} \cong 1,78 \text{ m/s}$$

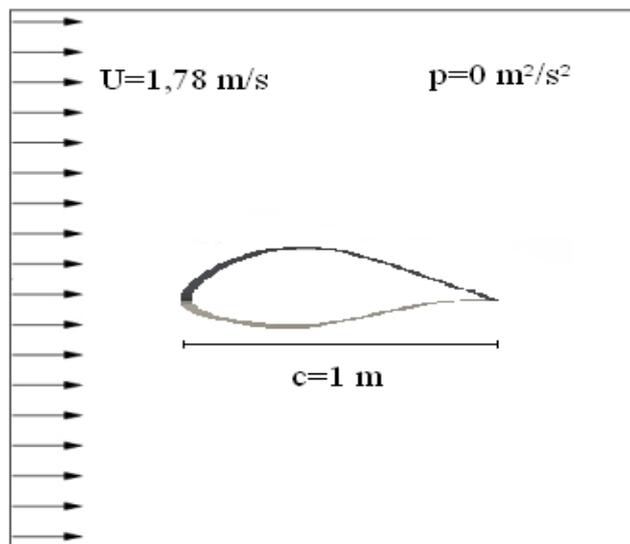


Figure 3.9 : Schémas du problème.

Ici l'écoulement est incompressible et la pression p est une pression cinétique (m^2/s^2), sa valeur absolue n'est pas très importante et c'est juste une pression de référence. Pour notre cas, cette pression est estimée nulle au début des calculs.

3.3.2.2. Prétraitement

Dans ce chapitre le maillage va être généré par un programme sur « GNU Octave » (très similaire à Matlab) associé à un logiciel de génération de maillage « GmshMeshGenerator ». Ensuite le maillage va être importé vers OpenFOAM par la commande « mshToFoam ». Le programme « foilgmsh.m » [12] produit un maillage multi-bloc, structuré à partir d'un fichier « .dat ». L'avantage de cette solution est qu'on peut modifier l'angle d'incidence à partir d'un fichier des entrées « Execution ». On a aussi l'avantage de choisir les paramètres de raffinement de maillage à partir de ce fichier.

❖ Structure du cas

Le système des fichiers pour ce cas suit le principe choisi dans le chapitre 1. Il faut savoir que la méthode de génération du maillage, le solveur choisi, ainsi que le choix des paramètres à visualiser influent sur la structure du cas. Le fichier « Expérimental » contient les valeurs expérimentales des coefficients de trainée et de portance en fonction de l'angle d'attaque.

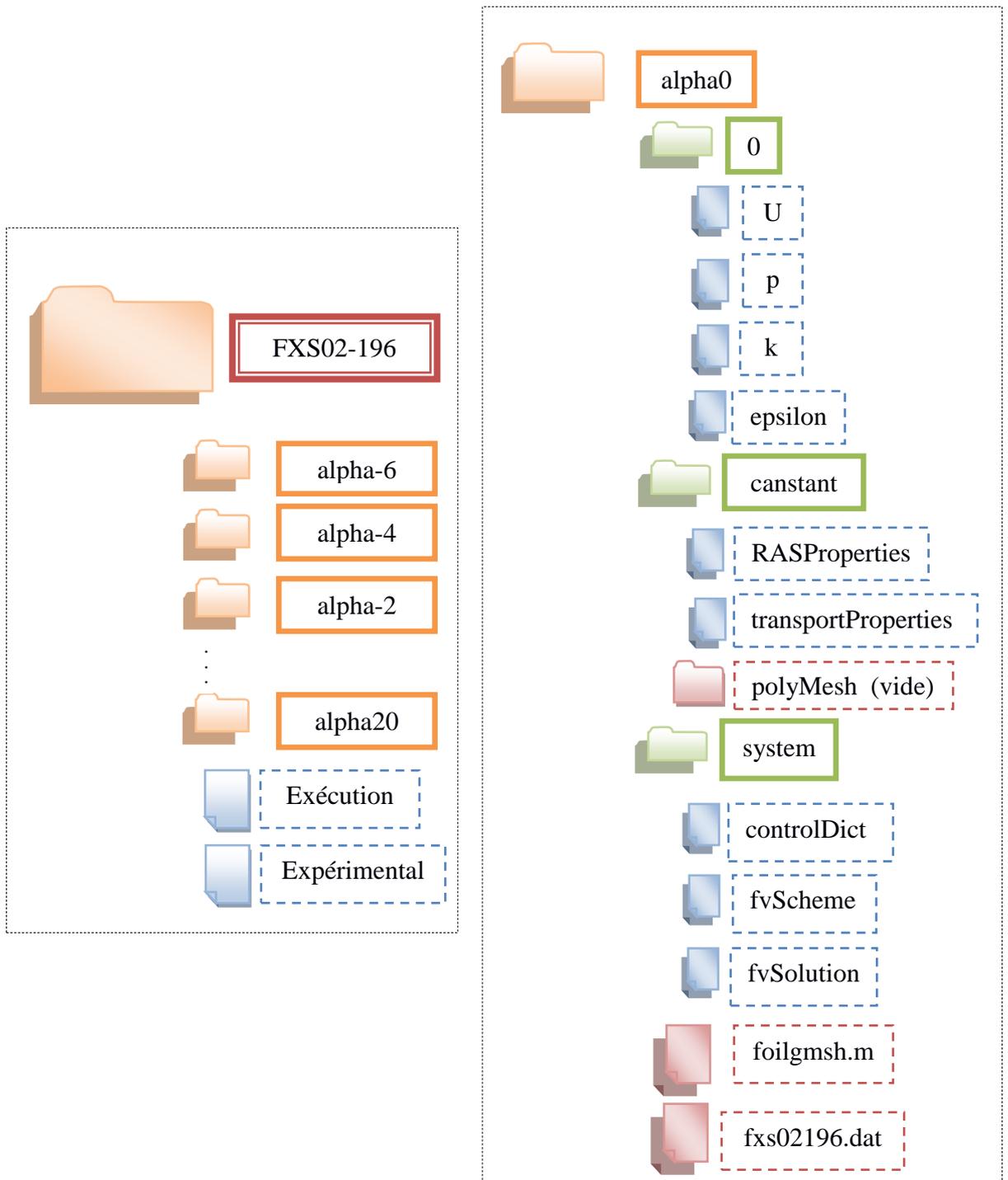


Figure 3.10 : Structure du cas FXS02196 sur OpenFOAM.

❖ Génération du maillage

On considère un écoulement visqueux autour d'un profil d'aile, c'est un problème très courant dans l'aérodynamique, le maillage nécessaire pour ce type de problème doit être raffiné dans les zones où le gradient de pression change subtilement. Le maillage nécessite d'être raffiné dans la couche limite du profil et le nombre adimensionnel y^+ doit être respecté tout en prenant compte du modèle de turbulence choisi. La distance minimale entre le profil et le domaine extérieur va influencer beaucoup sur les résultats. La méthode utilisée pour la création du maillage est très simple, grâce à un programme de génération de maillage, qui est créé spécialement pour le logiciel de maillage « Gmsh », et par l'intermédiaire de la commande « mshToFoam », on peut faire à partir d'un fichier de géométrie .DAT un maillage structuré autour d'un profil avec l'avantage de modifier l'angle d'incidence par un simple changement dans les entrées. C'est aussi là où un autre script « Exécution [A2] » entre en jeu pour modifier chaque fois l'angle et refaire le calcul. Le script est programmé aussi pour faire un assemblage des résultats à la fin de la simulation. Les logiciels utilisés sont tous des logiciels gratuits sous licence publique, et le fichier « foilgmsh.m »[12] est un travail personnel libre qui est téléchargeable et accepte tout type de changement.

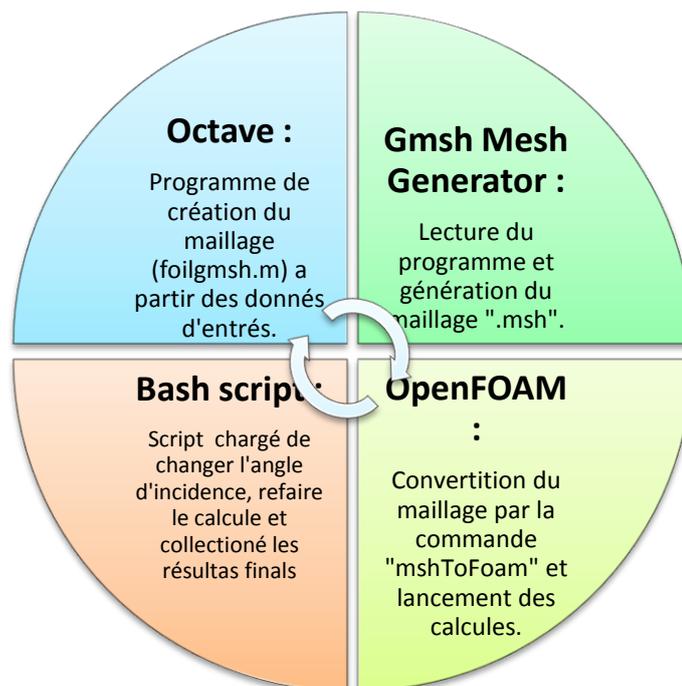


Figure 3.11 : La procédure suivie dans ce cas d'étude.

Le maillage obtenu est un maillage structuré hexagonale avec raffinement de la zone de couche limite ainsi que la zone de sillage derrière le profil. Le nombre des mailles est aussi important dans le bord d'attaque et les zones au-dessus et en dessous du profil car c'est là où le gradient de pression est très important.

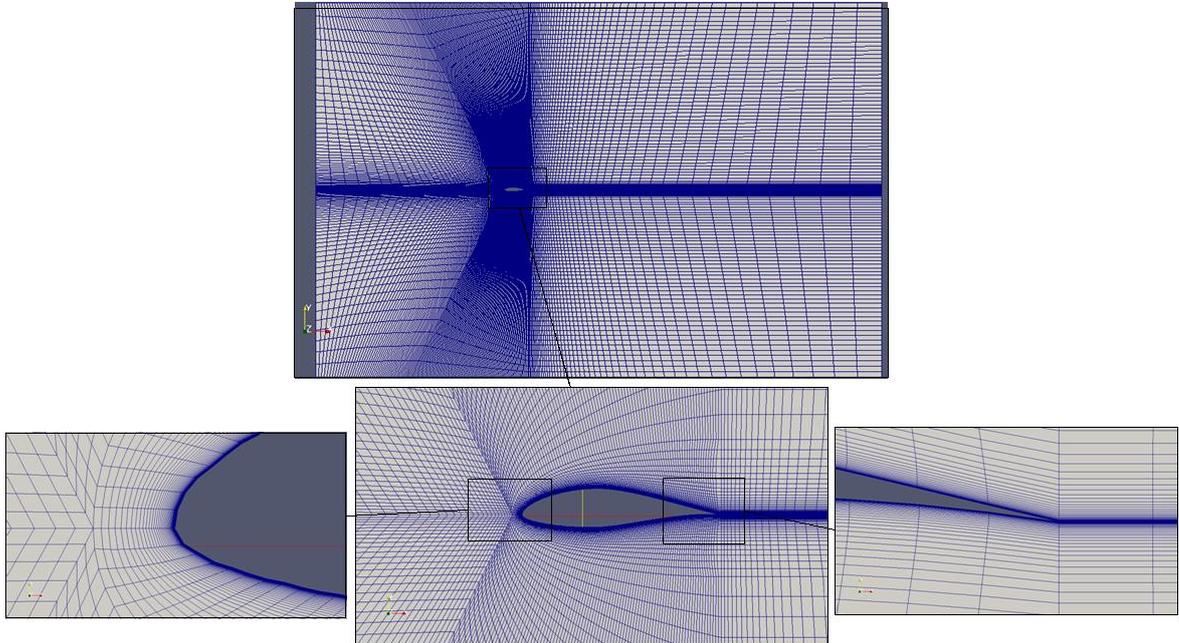


Figure 3.12: Visualisation du maillage à l'angle « 0 ».

3.3.2.3. Traitement

La partie de proceession, ou de traitement, consiste à faire le choix du solveur ainsi que les conditions aux limites pour la simulation. Cette étape n'est pas indépendante des autres, à vrai dire, le choix du maillage est justifié selon ce solveur. C'est ainsi que les régions de raffinement sont choisies.

❖ **Choix du solveur**

Le solveur utilisé est « simpleFoam », destiné au traitement des écoulements incompressibles stationnaires ($\frac{\partial}{\partial t} = 0$) et le modèle choisi pour l'évaluation de la turbulence est le modèle a deux équations « *k—ε standard* ». La simulation est établie pour un angle de 0°, Ensuite, grâce a un script, on ordonne l'ordinateur de changer l'angle, faire un nouveau maillage et lancer le calcul une fois de plus. La procédure se répète dans une boucle pour 19 angles d'incidences, enfin, les résultats sont regroupés et le tracer se fait à l'aide du logiciel Gnuplot.

❖ Conditions aux limites

Pour les conditions limites du problème, la masse volumique du fluide sera $\rho = 1,225 \text{ kg/m}^3$ et sa viscosité $\nu = 1,46 \times 10^{-5} \text{ kg/m.s}$. La corde du profil est égale à l'unité $C = 1 \text{ m}$ et la vitesse d'écoulement est $U = 1,7827 \text{ m/s}$.

$$Re = \frac{1 \times 1,7827}{1,46 \times 10^{-5}} = 122102,7397$$

Tableau 3.3 : Conditions aux limites.

Paramètres	Faces	Conditions aux limites	Description
U	inlet	fixedValue, uniform(1,7827 0 0)	Vitesse uniforme de 1,7827m/s selon l'axe des x.
	outlet	inletOutlet	Écoulement de sortie.
	airfoil	fixedValue, uniform (0 0 0)	Vitesse nulle (condition de non glissement).
	top	symmetryPlane	Plan de symétrie (domaine infini).
	bottom	symmetryPlane	//
	frontAndBack	Empty	Faces vides (bidimensionnel).
P	Inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	Outlet	fixedValue, uniform 0	Estimation de la pression à la sortie.
	airfoil	zeroGradient	Gradient nulle dans la direction de la normale.
	top	symmetryPlane	Plan de symétrie (domaine infini).
	bottom	symmetryPlane	//
	frontAndBack	Empty	Faces vides.
k	inlet	fixedValue, uniform $7,67e - 5$	Estimation de k à l'entrée.
	outlet	inletOutlet	Écoulement de sortie.
	airfoil	kqRWallfunction, uniform 0	Fonction d'évaluation de k dans la couche limite.
	top	symmetryPlane	Plan de symétrie (domaine infini).
	bottom	symmetryPlane	//
	frontAndBack	Empty	Faces vides (bidimensionnel).
Epsilon	inlet	fixedValue, uniform $3,02e - 5$	Vitesse uniforme 1,7842m/s fixe.
	outlet	inletOutlet	Gradient nulle dans la direction de la normale.
	airfoil	epsilonWallfunction, uniform 0	Vitesse nulle (condition de non glissement).
	top	symmetryPlane	Plan de symétrie (domaine infini).
	bottom	symmetryPlane	//
	frontAndBack	Empty	Faces vides (bidimensionnel).

3.3.2.4. Post-traitement

Dans cette partie les résultats obtenus sont automatiquement adaptés pour être visualisé sous Paraview, le logiciel de post-traitement standard d'OpenFOAM. Le calcul des coefficients de portance et de trainé est obtenu en introduisant une fonction dans le fichier « controlDict » situé dans le dossier « system ». Les résultats seront ensuite récapitulés dans un fichier puis, à l'aide de Gnuplot, on trace les courbes demandées.

```

application    simpleFoam;    // Le solveur utiliser.
startFromlatestTime; // Début a partir d'Une valeur« 0 »,« 0.13 », « 1000 »
startTime0;    // ou a partir de la dernier valeur qui existe« latestTime ».
stopAtendTime; // Le temps d'arrêt.
endTime2500;
deltaT1; // Le pas de temps  $\Delta T$  (ici le temps est juste un paramètre d'itération (stationnaire).
writeControltimeStep; // Paramètres d'écriture et des sorties.
writeInterval2500;
purgeWrite0;
writeFormatascii;
writePrecision7;
writeCompression off;
timeFormat    general;
timePrecision6;
runTimeModifiabletrue;
functions      // Ensemble des fonctions auxiliaires.
(
forceCoeffs    // Les coefficients de force (portance, trainé et moments).
{
typeforceCoeffs;
functionObjectLibs( "libforces.so" ); // Emplacement du dictionnaire.
    patches      ( "airfoil" ); // Le nom de la paroi.
pName          p;
UName          U;
rhoNamerhoInf;
rhoInf1.225;
CofR           (0.245 -0.068 0); // Le centre de rotation (pour le coefficient de moment).
liftDir        (0 1 0); // Le vecteur unitaire de la portance.
dragDir        (1 0 0); // Le vecteur unitaire de la trainée.
pitchAxis      (0 0 1); // L'axe tangage.
magUInf43.8; // Le module de U.
lRef1; // La corde de référence.
Aref0.1; // La surface de référence.
outputControltimeStep; // Paramètres des sorties.
outputInterval10;
}
);

```

Les graphs obtenus ont une allure satisfaisante, le C_z est presque linéaire entre -6° et 8° le profil de FXS02196 a un angle de décrochage très grand (Figure3.13) mais sa grande épaisseur engendre une grande trainée même dans les faibles incidences (Figure3.14).

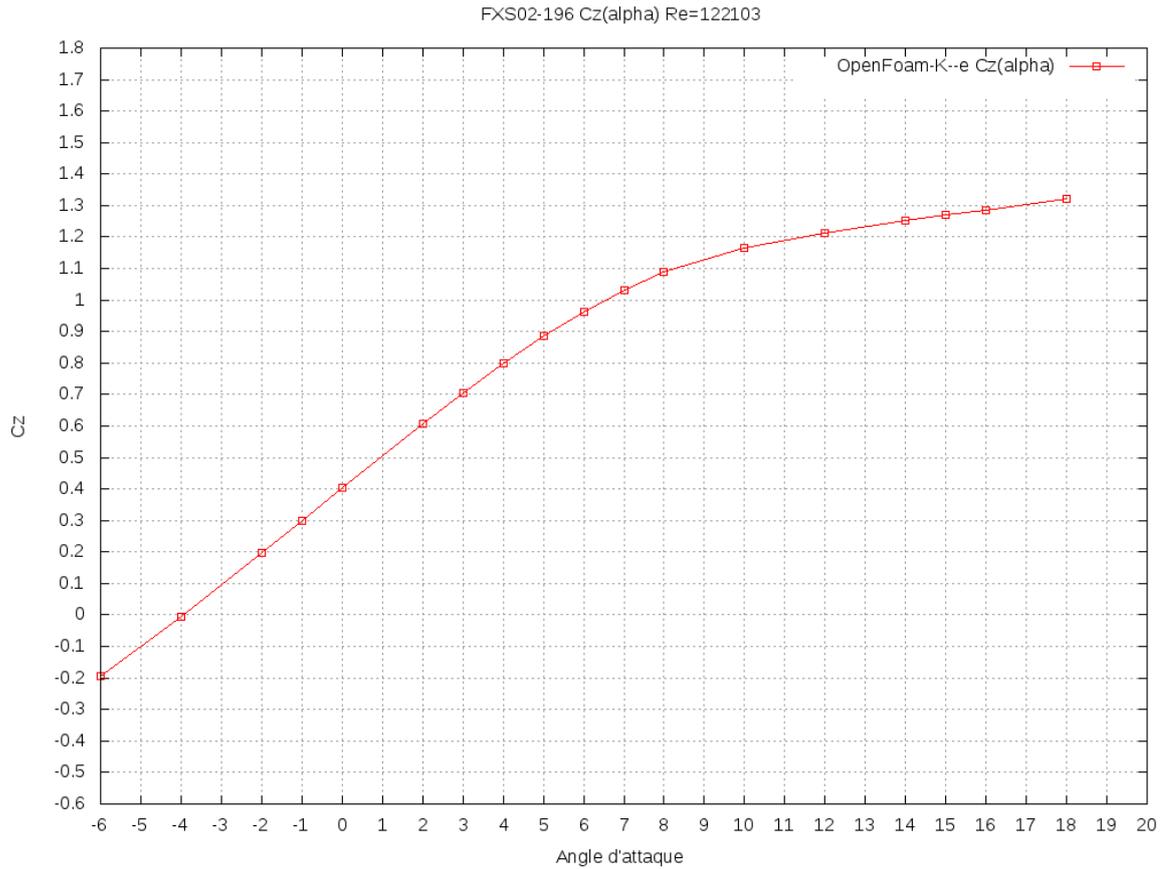


Figure 3.13 : Cz en fonction de l'angle d'attaque.

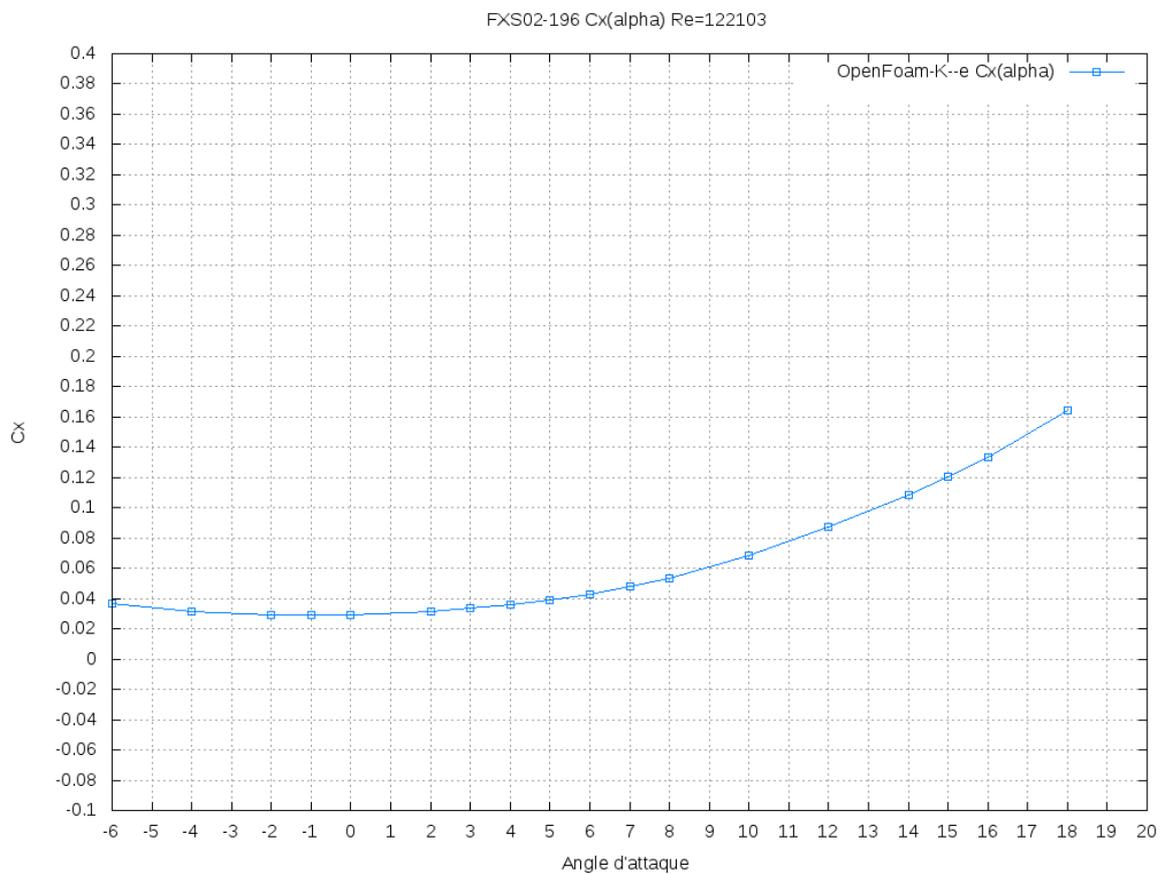


Figure 3.14 : Cx en fonction de l'angle d'attaque.

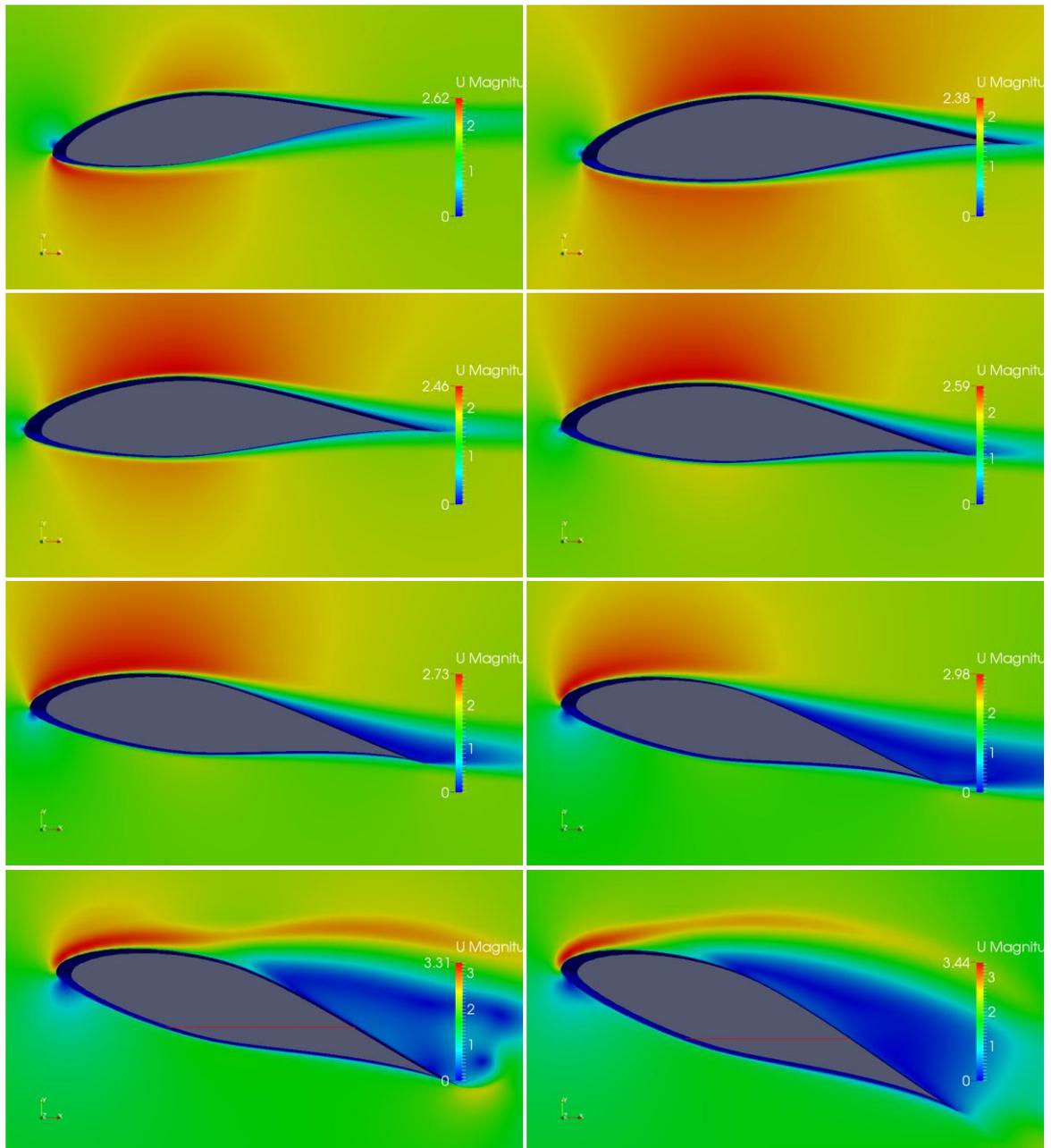


Figure 3.15 : Visualisation du champ de vitesse pour des angles d'attaque différents (-6, -2, 0, 4, 8, 12, 16, 20).

La figure 3.15 ci-dessus montre l'évolution des champs de vitesse en fonction de l'angle d'attaque (de -2° à 20°)

3.3.2.5. Comparaison des Résultats

Les résultats obtenus sont comparés avec ceux trouvés dans l'expérimental, on peut voir que les deux graphs suivent la même allure mais avec quelques erreurs, Ces erreurs sont dues à des imperfections dans l'expérience (perturbations dans la salle des essais, la finition du profil, erreurs des instruments) et au modèle de turbulence choisit qui peut être délicat à choisir dans cette plage de Reynolds.

Il faut aussi savoir qu'à des faibles nombres de Reynolds, il se peut que des bulles de séparation apparaissent à la fin de l'extrados ce qui rend le C_x à prédire. On ce qui concerne le C_z , les valeurs obtenus par OpenFOAM doivent être proche de la réalité car son calcul dérive du calcul des forces de pression contrairement au C_x qui dépend fortement des forces de frottement.

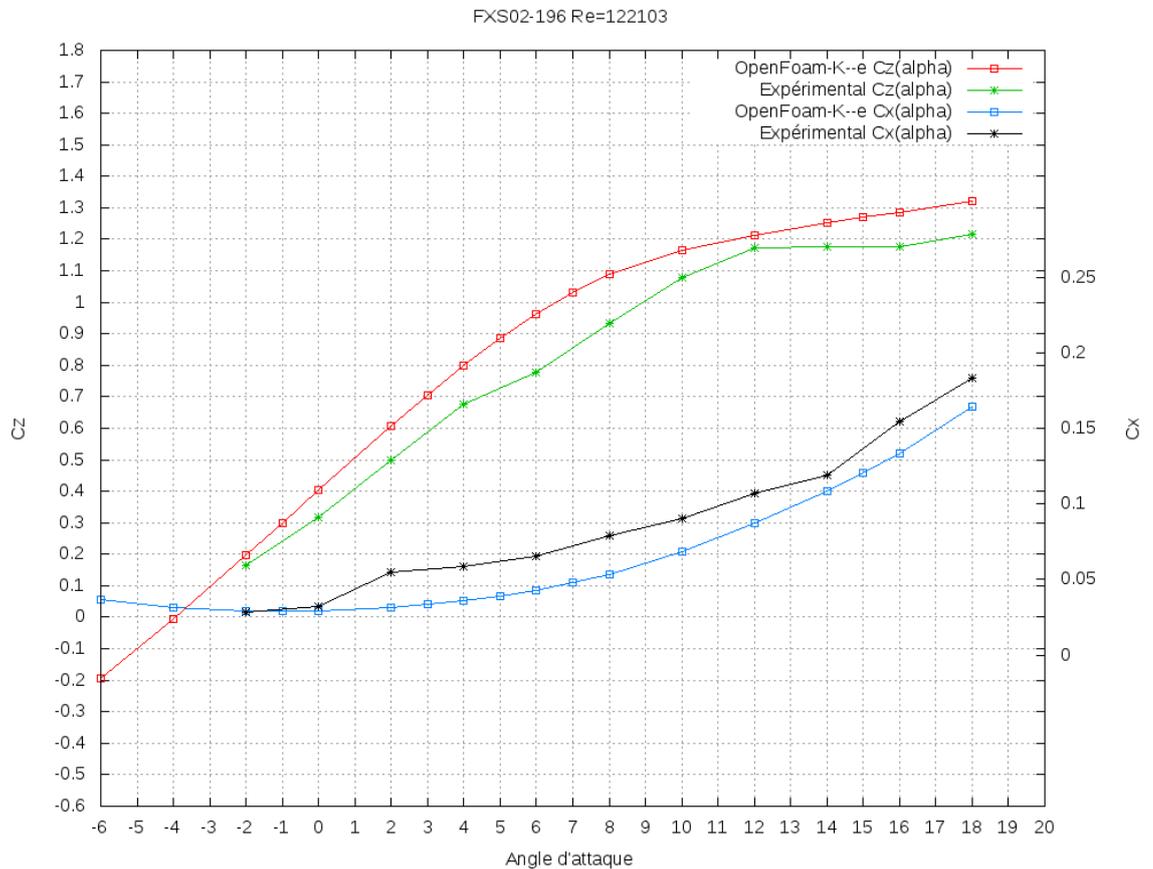


Figure 3.16 : Comparaison de C_z et C_x en fonction de l'angle d'attaque

3.4. Le profil NACA4412

Il s'agit d'un profil NACA4412, c'est un profil de NASA conçu pour évoluer à différents nombre de Reynolds, c'est aussi un profil d'intrados plat (flat bottom). Grace à sa simplicité et ces bonnes performances aérodynamiques, il est un des choix préférés des constructeurs. Dans ce chapitre on va utiliser OpenFOAM pour déterminer les coefficients aérodynamiques du profil d'aile NACA4412 à un nombre de Reynolds bien précis et comparer les résultats avec celle de la littérature afin de montrer comment se servir d'OpenFOAM pour faire un des calculs de base dans l'aérodynamique.

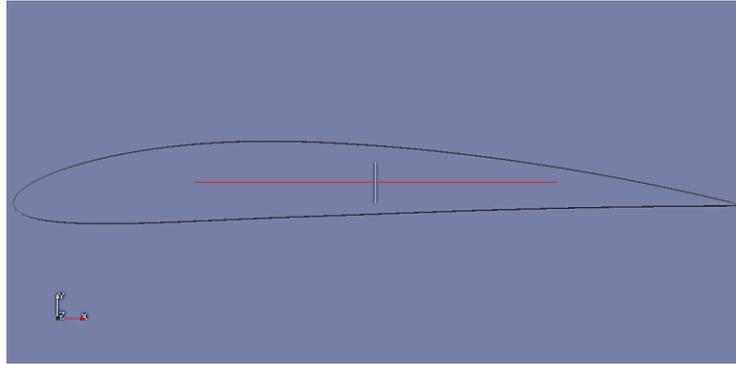


Figure 3.17 : Profil NACA4412.

3.4.1. Présentation

Il s'agit d'un écoulement incompressible ($M \ll 0,3$), bidimensionnel autour du profil NACA4412. Le nombre de Reynolds est égal à 3000000.

Hypothèses :

- Écoulement incompressible.
- Écoulement newtonien.
- Écoulement bidimensionnel.
- Effets de gravité négligeables.

Le fluide considéré est toujours l'air ambiant :

$$\rho = 1,225 \text{ kg/m}^3$$

$$\vartheta = 1,46 \times 10^{-5} \text{ kg/m.s}$$

La corde du profil est égale à l'unité $C = 1 \text{ m}$ et la vitesse d'écoulement est

$$U = \frac{Re \times \vartheta}{C} = \frac{3000000 \times 1,46 \times 10^{-5}}{1} \cong 43,8 \text{ m/s}$$

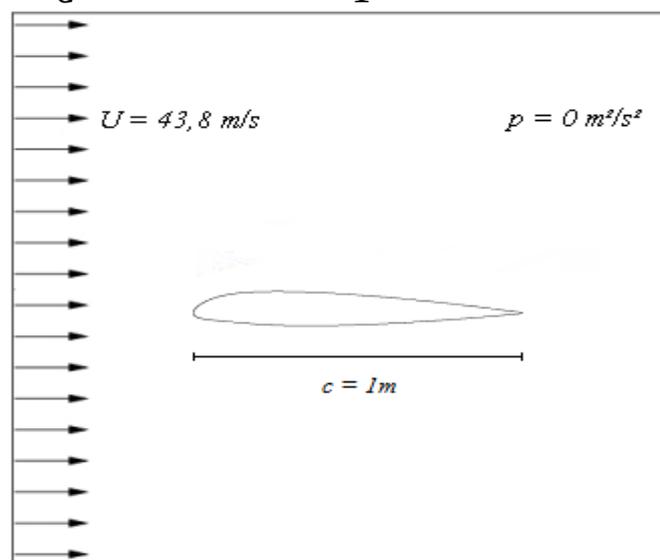


Figure 3.18 : Schémas du problème.

3.4.2. Prétraitement

Les procédures du prétraitement sont identiques à celles du profil précédent sauf pour les conditions aux limites et le model de turbulence choisi.

❖ Structure du cas

Le système des fichiers pour ce cas est très semblable au cas précédent :

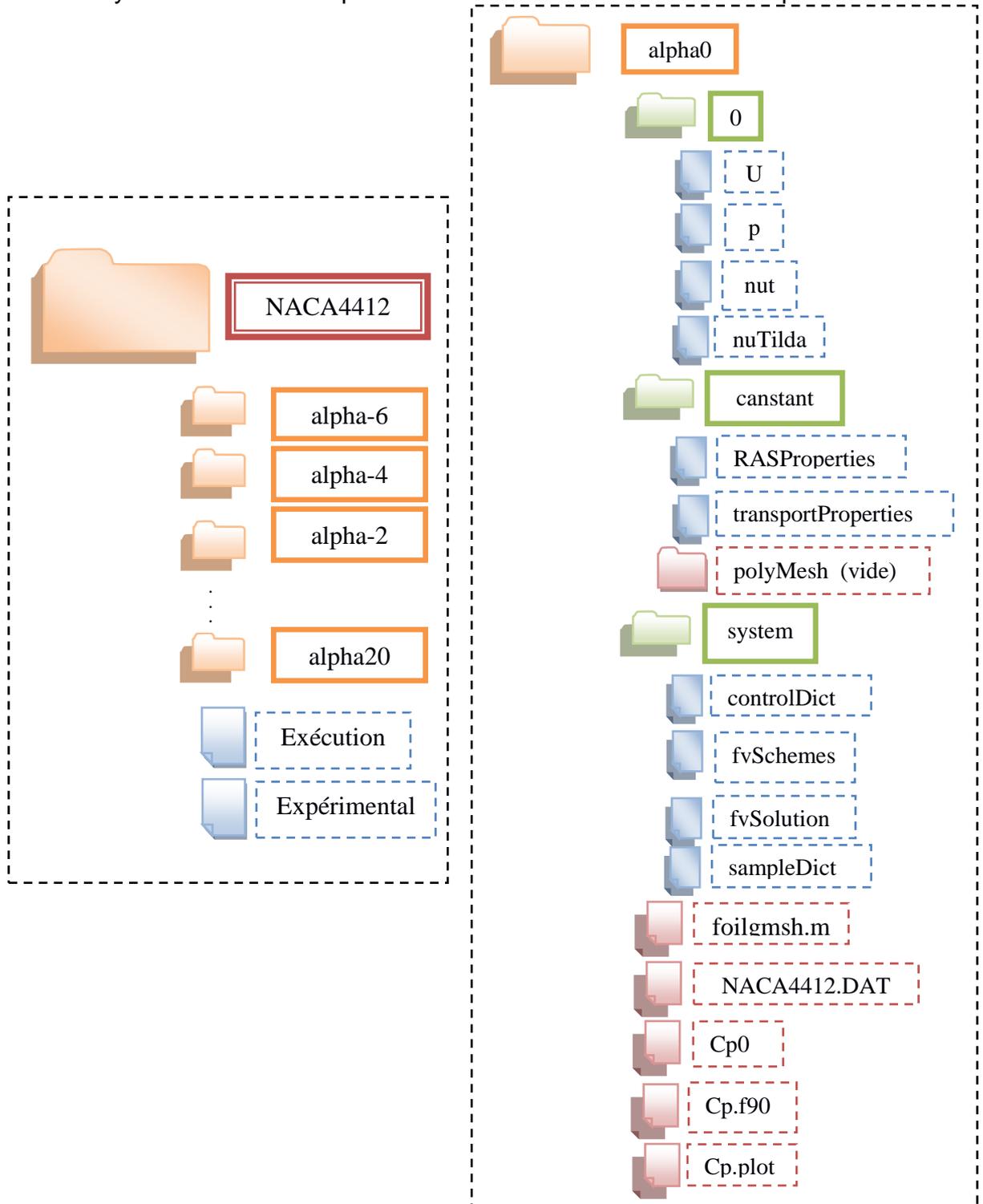


Figure 3.19 : Structure du cas NACA4412 sur OpenFOAM.

Le fichier « Cp0 » correspond au coefficient de pression expérimental à l'angle d'incidence 0. Le fichier Cp.f90 est un petit programme en Fortran pour organiser dans des vecteurs les valeurs de « p » qui provient de la commande « sample » et calcule le coefficient de pression à chaque point du profil. Enfin, le fichier « Cp.plot » trace les graphs obtenus avec Gnuplot.

❖ Génération du maillage

La procédure suivie pour la génération du maillage est la même que celle montrée dans la figure 3.11. Le maillage est plus raffiné dans la couche limite afin de respecter le y^+ qui correspond au type d'écoulement. Chaque angle d'incidence est étudié à part puis un script refait les calculs pour les angles voulus. Le maillage obtenu est un maillage structuré hexagonale avec raffinement de la zone de couche limite ainsi que la zone de sillage derrière le profil. Le nombre des mailles est aussi important dans le bord d'attaque et les zones au-dessus et en dessous du profil ou le gradient de pression est très important.

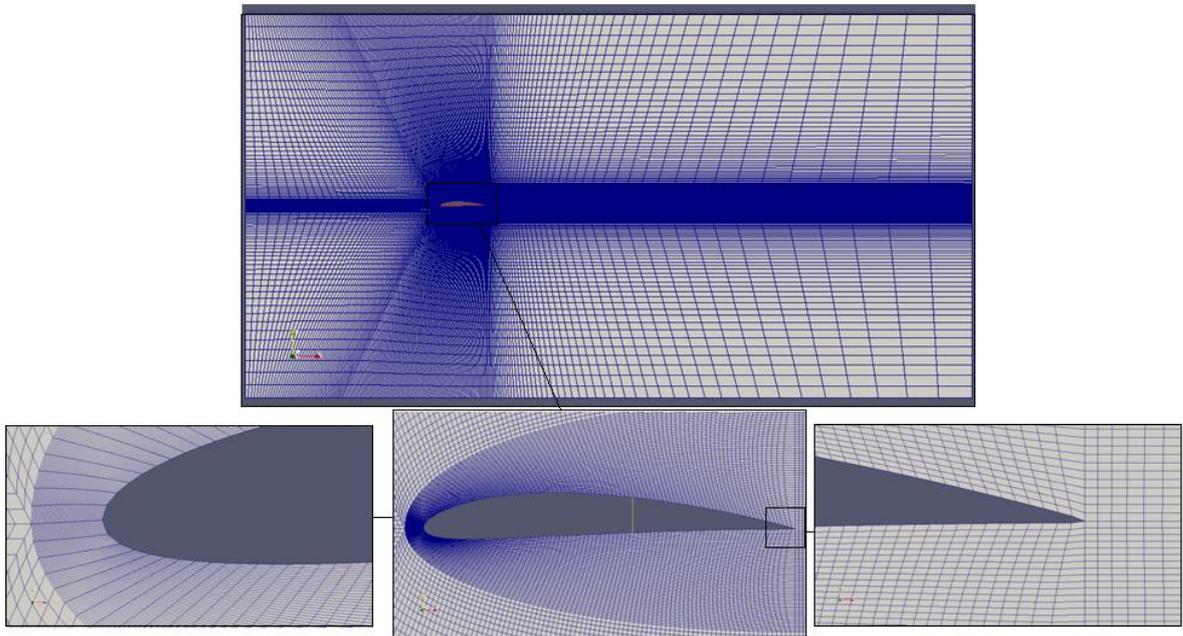


Figure 3.20: Visualisation du maillage a l'incidence « 0 ».

3.4.3. Traitement

Le solveur choisit est le même que celui choisi pour le cas précédant puisqu'il s'agit du même type d'écoulement sauf que le nombre de Reynolds est plus élevé et par conséquent le model de turbulence est différent.

❖ Choix du solveur

Le solveur utilisé est « simpleFoam », destiné aux écoulements incompressibles stationnaires et le modèle choisi pour l'évaluation de la turbulence est le modèle « SpalartAllmaras ». La simulation est établie pour un angle de 0° , Ensuite, grâce à un script on ordonne l'ordinateur de changer l'angle, faire un nouveau maillage et lancer le calcul une fois de plus. La procédure se répète dans une boucle pour 19 angles d'incidence puis, enfin, les résultats sont regroupés et le tracer se fait à l'aide de Gnuplot.

❖ Conditions aux limites

Pour les conditions limites du problème, la masse volumique du fluide sera $\rho = 1,225 \frac{kg}{m^3}$ et sa viscosité $\nu = 1,46 \times 10^{-5} kg/m.s$. La corde du profil est $C = 1 m$ et la vitesse d'écoulement est

$$U = 43,8 m/s.$$

$$Re = \frac{1 \times 43,8}{1,46 \times 10^{-5}} = 3000000$$

Tableau 3.4 : Conditions aux limites.

Paramètres	Faces	Conditions aux limites	Description
U	inlet	fixedValue, uniform(43,8 0 0)	Vitesse uniforme 43,8m/s selon l'axe des x.
	outlet	inletOutlet	Ecoulement de sortie libre.
	airfoil	fixedValue, uniform (0 0 0)	Vitesse nulle (condition de non glissement).
	top	symmetryPlane	Plan de symétrie (domaine infini).
	bottom	symmetryPlane	//
	frontAndBack	Empty	Faces vides (bidimensionnel).
P	inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	outlet	fixedValue, uniform 0	Estimation de la pression a la sortie.
	airfoil	zeroGradient	Gradient de pression nulle
	top	symmetryPlane	Plan de symétrie (domaine infini)
	bottom	symmetryPlane	//

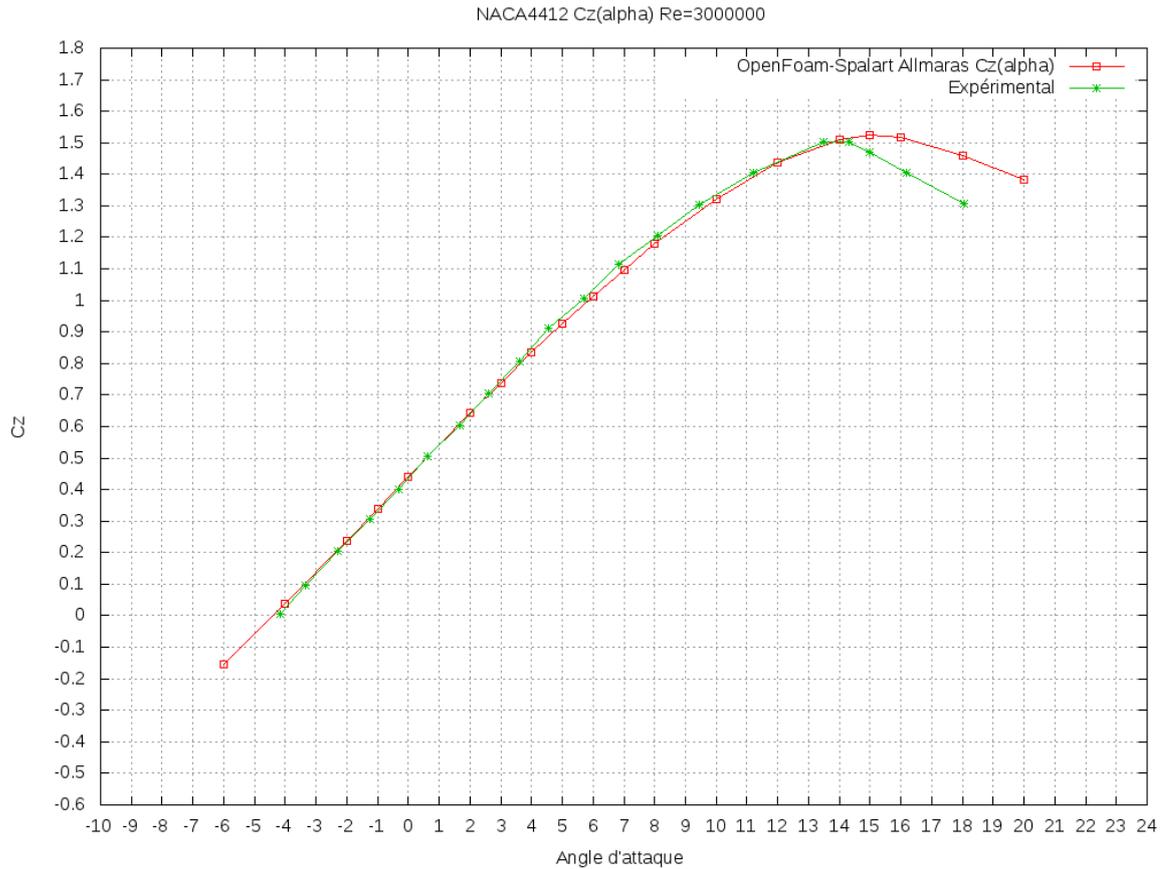
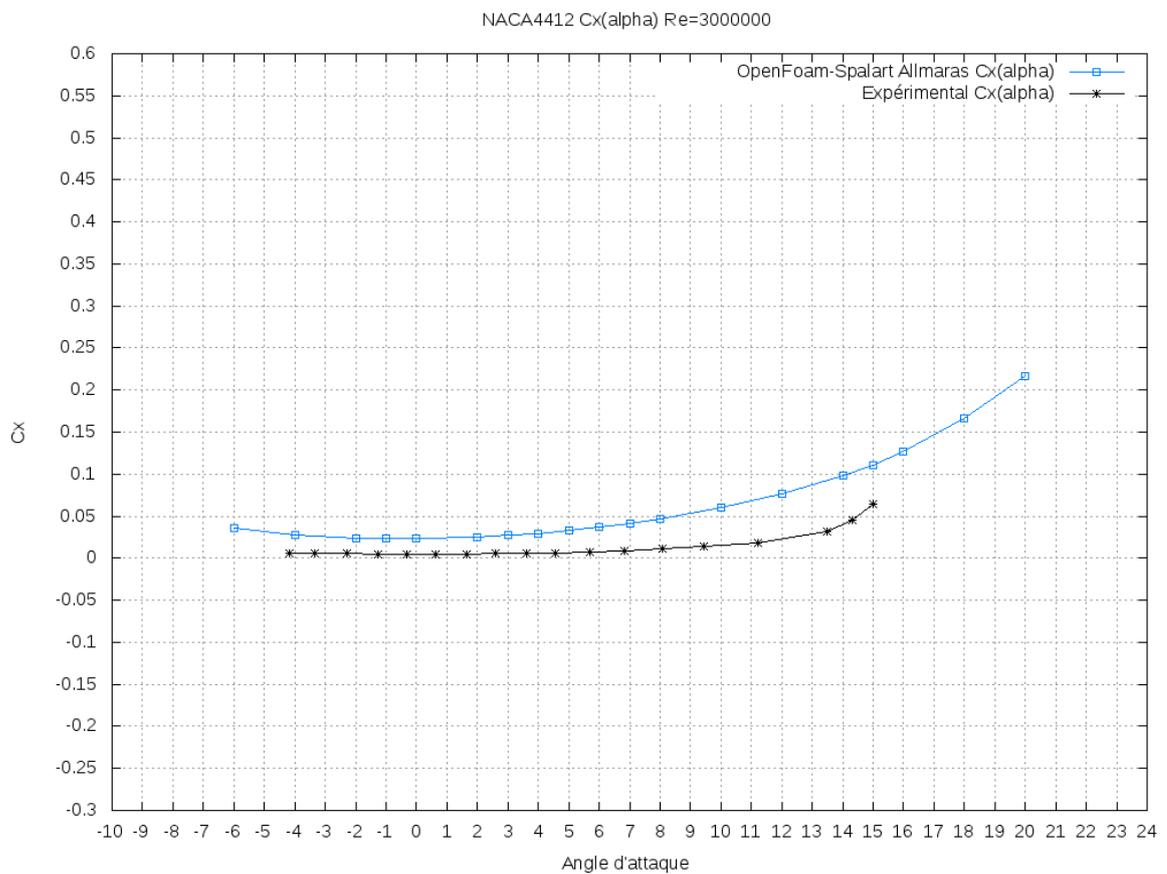
	frontAndBack	Empty	Faces vides
nut	inlet	freestream, uniform0,0385	Estimation de ϑ_t a l'entré
	outlet	freestream, uniform0,0385	Estimation de ϑ_t a la sortie
	airfoil	nutUSpladingWallFunction, uniform 0	Fonction d'évaluation de ϑ_t dans la couche limite
	top	symmetryPlane	Plan de symétrie (domaine infini)
	bottom	symmetryPlane	//
	frontAndBack	empty	Faces vides (bidimensionnel)
nuTilda	inlet	freestream, uniform0,1929	Estimation de $\tilde{\vartheta}$ a l'entré
	outlet	freestream, uniform01929	Estimation de $\tilde{\vartheta}$ a la sortie
	airfoil	fixedValue, uniform 0	Estimation de $\tilde{\vartheta}$ dans la couche limite
	top	symmetryPlane	Plan de symétrie (domaine infini)
	bottom	symmetryPlane	//
	frontAndBack	empty	Faces vides (bidimensionnel)

3.4.4. Post-traitement

Dans cette partie, Les résultats obtenus sont automatiquement adapté pour être visualisés grâce à Paraview, le logiciel de post-traitement standard d'OpenFOAM. Le calcul des coefficients de portance et de trainé est obtenu en introduisant une fonction dans le fichier « controlDict » pareille au cas précédant. Tandis que pour le calcul de C_p , il n'existe pas une fonction dans OpenFOAM du coefficient de pression. Alors, on introduit la commande « sample » avec son fichier d'instruction sampleDict situé dans le dossier « system ».

➤ **Coefficients de trainé et de portance**

Les courbes de traînée et de portance ont été tracées pour des incidences allant de -6 à 20 degrés. Les graphs obtenus suivent la même allure que celle de l'expérimental, le profil de NACA4412 à un angle de décrochage d'environ 15° pour ce nombre de Reynolds (Figure 3.21). Cependant comme on pouvait s'y attendre, ces calculs stationnaires rencontrent des problèmes de convergences à partir de l'angle de décrochage. On retrouve ci-dessous (Figure 3.23) l'évolution de la polaire $C_z(C_x)$ au cours de la simulation pour différents angles.

Figure 3.21 : Comparaison de C_z en fonction d'angle d'attaque.Figure 3.22 : Comparaison de C_x en fonction d'angle d'attaque.

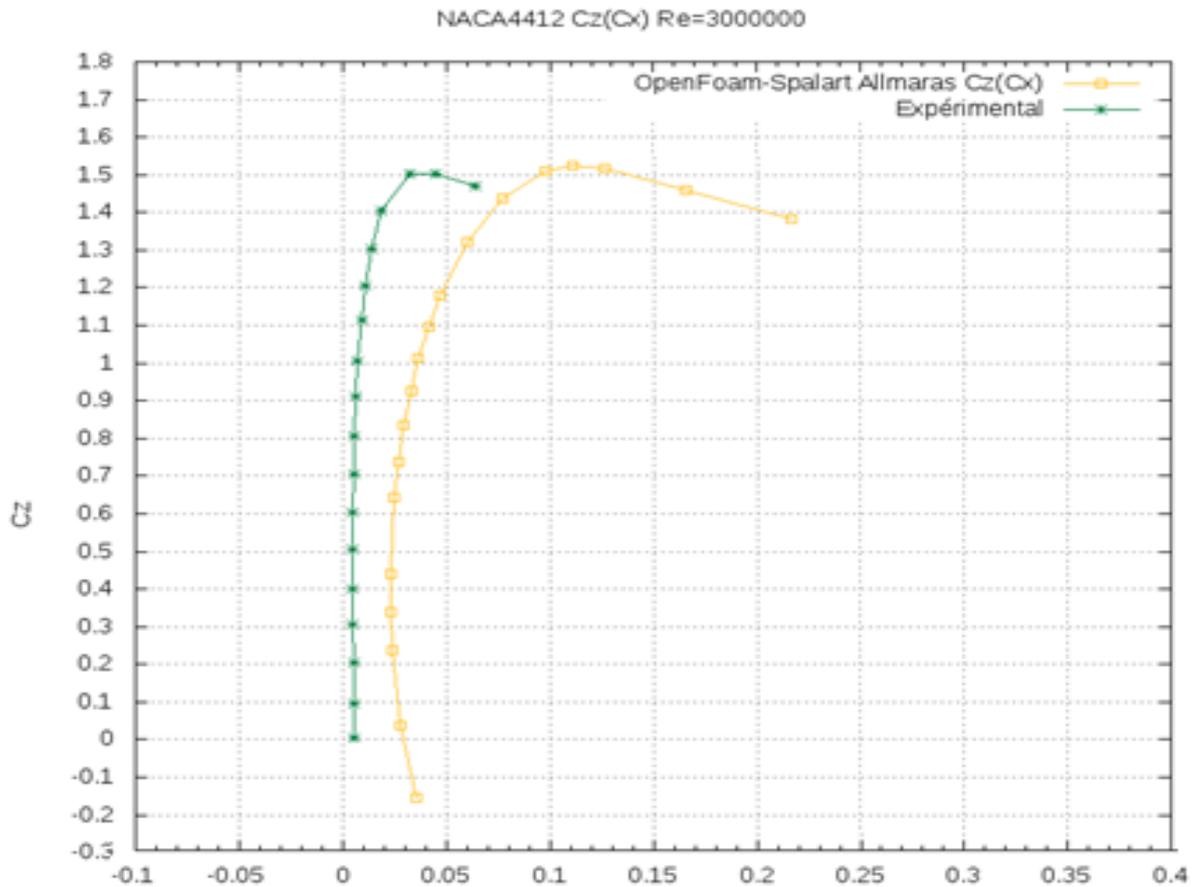


Figure 3.23 : Comparaison de la polaire.

➤ Coefficients de pression

Avec la commande « sample » on peut obtenir la valeur d'un champ scalaire ou vectoriel dans n'importe quelle région sous forme de fichiers de sorties. On peut ainsi trouver la pression à n'importe quelle point de la région indiquée, ensuite, par un simple programme de FORTRAN, qui peut lire les sorties de cette commande, on peut calculer la pression moyenne à l'infini et par conséquent le coefficient de pression pour chaque point du profil. Dans les figures suivantes on s'intéresse à l'évolution du coefficient de pression C_p à l'intrados et à l'extrados du profil en fonction de la corde adimensionnel X/c . Cette évolution est représentée sur les figures ci-dessous. La courbe supérieure se réfère à l'extrados et la courbe inférieure se réfère à l'intrados. À l'extrados, on met clairement en évidence la forte dépression près du bord d'attaque du profil. Pour ce qui est de l'intrados, on observe aussi une dépression près du bord d'attaque, mais bien moins importante que celle de l'extrados. En comparant avec les valeurs expérimentales, on remarque que le C_p a presque la même allure.

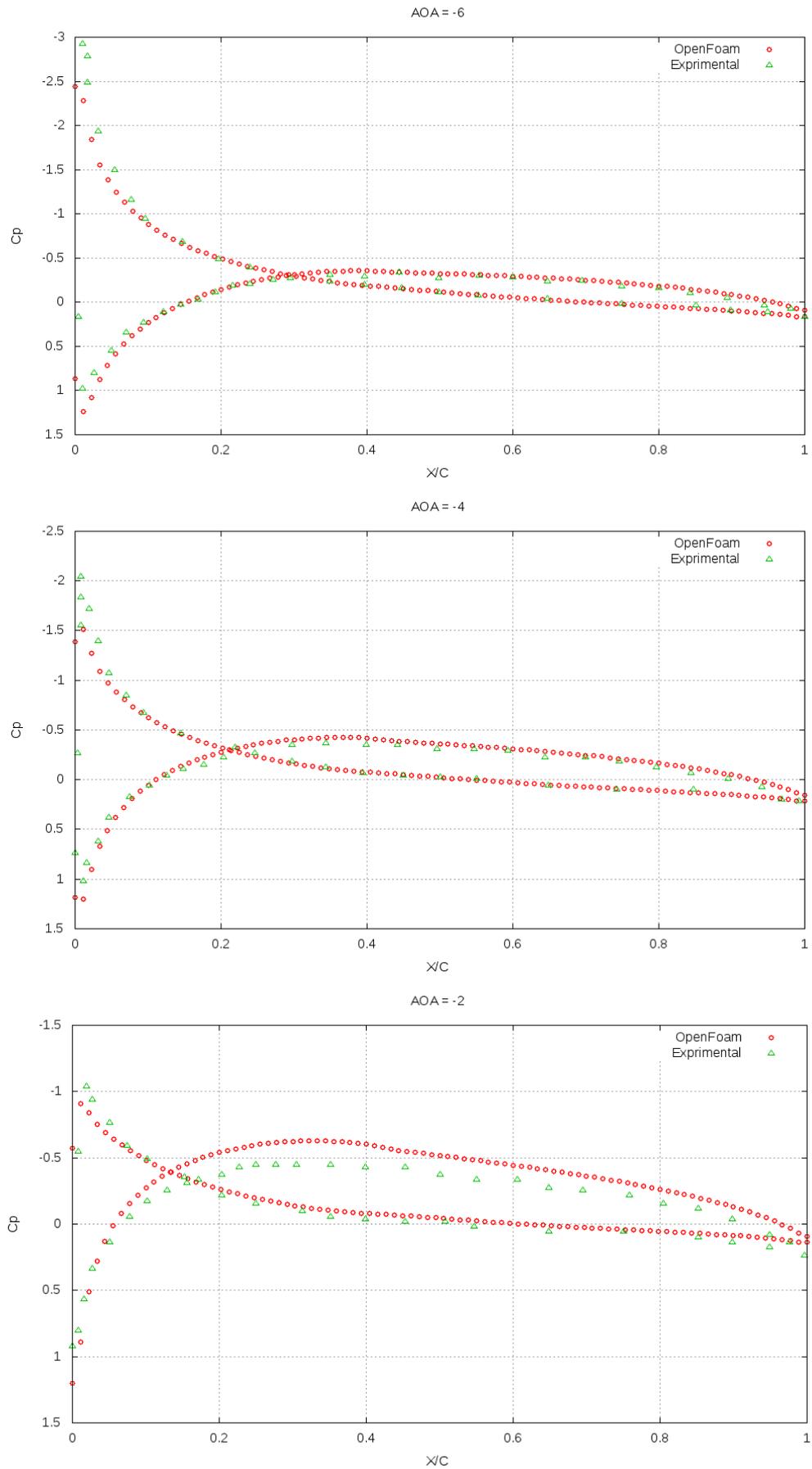


Figure 3.24 : Evolution de C_p en fonction de la corde pour $\alpha = -6^\circ, -4^\circ$ et -2° .

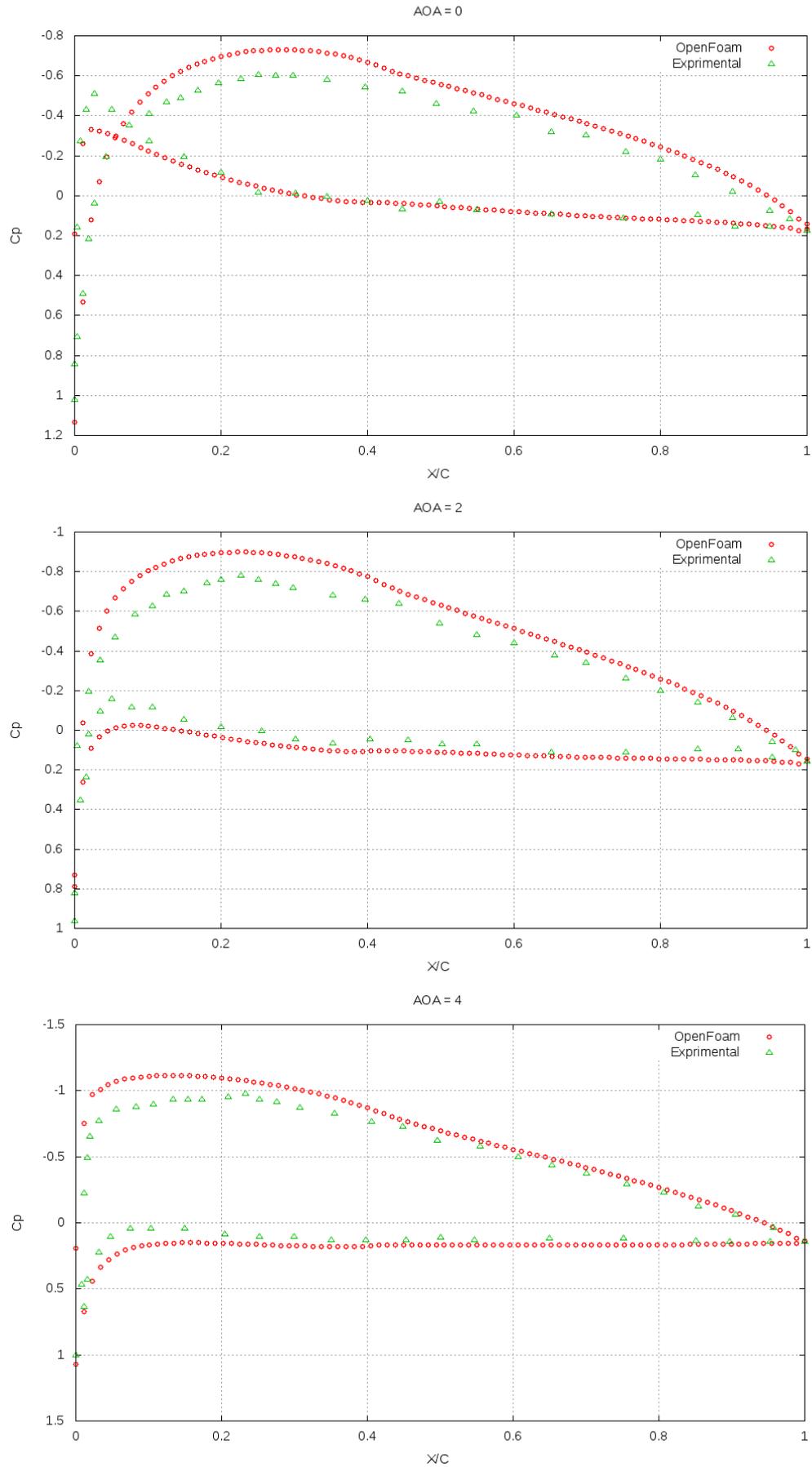


Figure 3.25 : Evolution de C_p en fonction de la corde pour $\alpha = 0^\circ, 2$ et 4° .

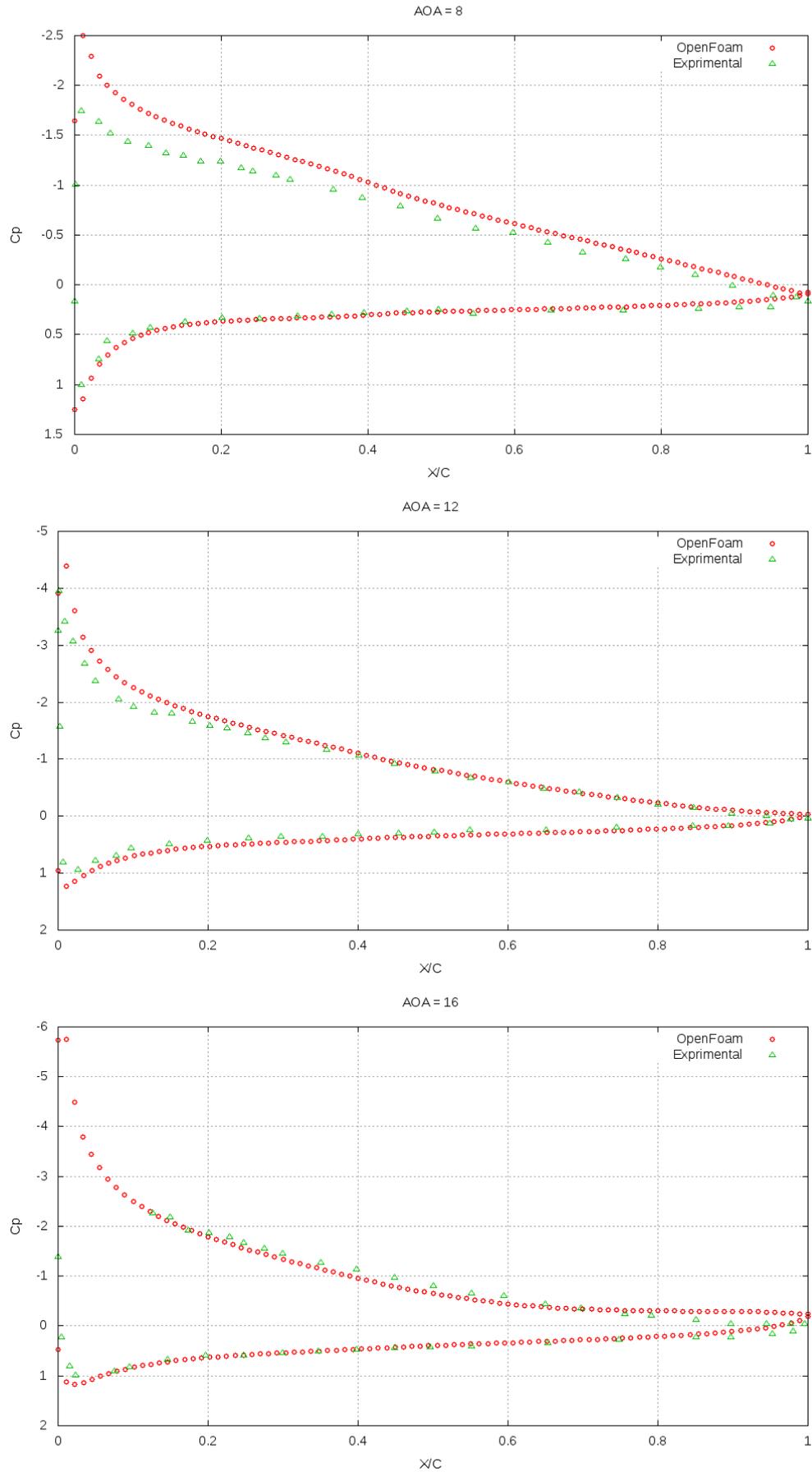


Figure 3.26 : Evolution de C_p en fonction de la corde pour $\alpha 8^\circ, 12^\circ$ et 16° .

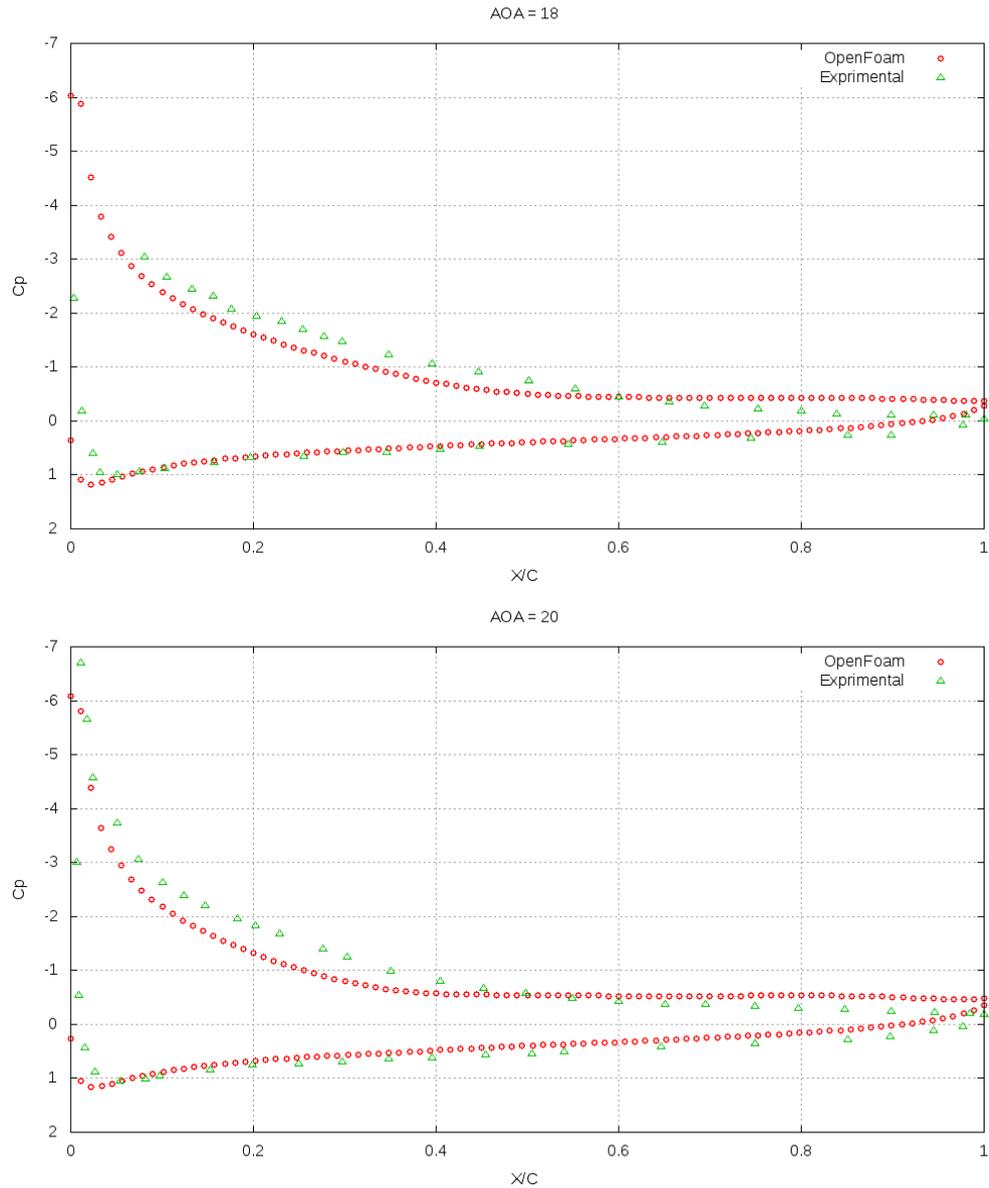


Figure 3.27 : Evolution de C_p en fonction de la corde pour $\alpha = 18^\circ$ et 20° .

On voit que les courbes obtenues confirment très bien avec l'expérimental[11]. Les deux courbes suivent la même évolution en fonction de l'angle d'attaque et de la corde adimensionnel. On note aussi que les erreurs sont maximales dans le premier quart de la corde.

➤ Evolution du champ de vitesse

La figure 3.28 ci-dessous montre l'évolution des champs de vitesse en fonction de l'angle d'attaque (de -6° à 20°).

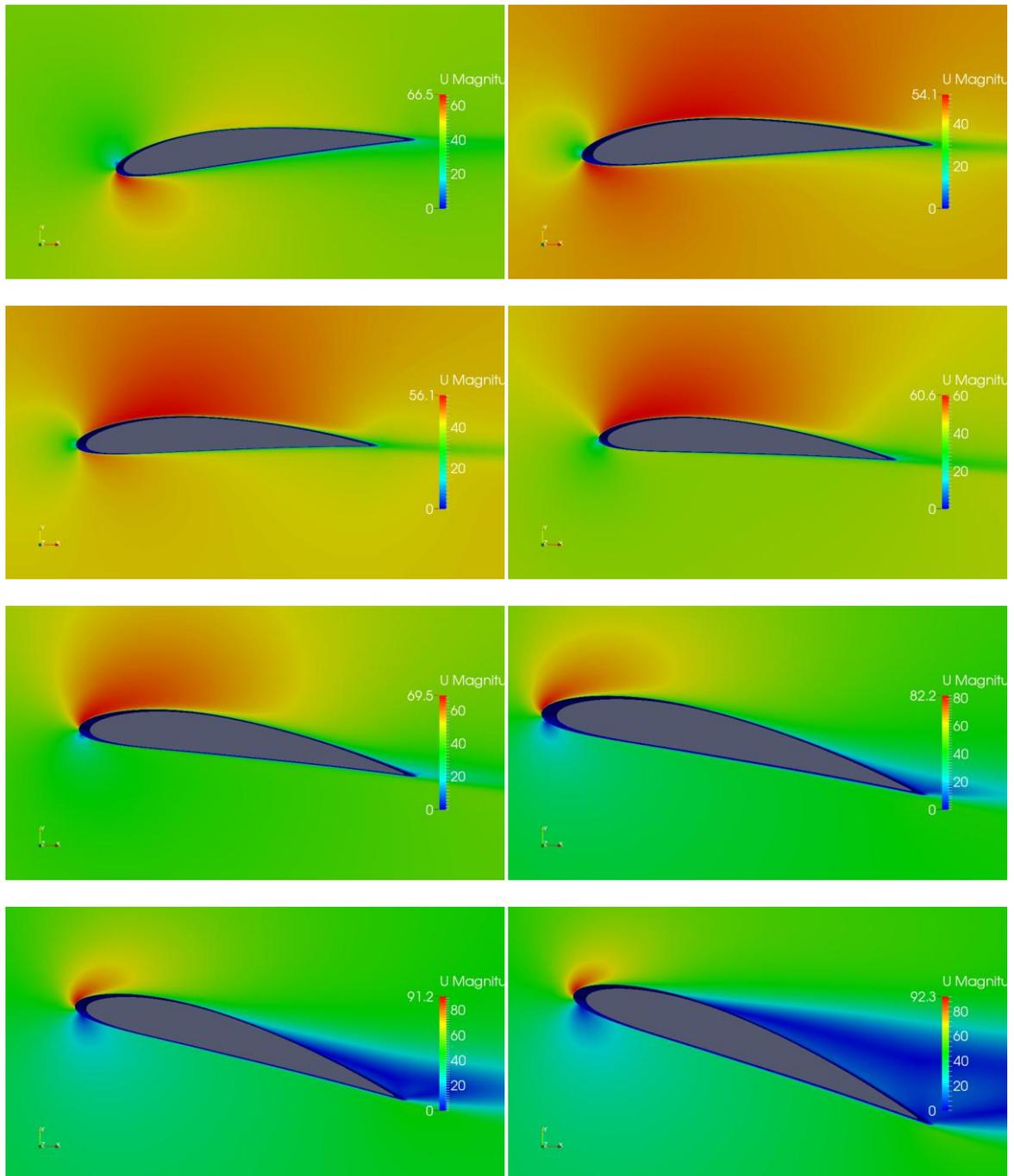


Figure 3.28 : Visualisation du champ de vitesse a différent angle d'attaque (-6, -2, 0, 4, 8, 12, 16, 20).

On voit que pour de faibles angles d'incidence la vitesse est très importante à l'extrados ce qui annonce une grande dépression. Quand l'angle d'incidence augmente, la valeur maximale de la vitesse augmente aussi et la couche limite se détache progressivement de la paroi.

3.5. Conclusion

En comparaison avec la référence, les résultats semblent plus proches de la réalité. La chute de portance est bien transcrite et l'angle de décrochage est proche de l'expérimental. On remarque que la traînée est surestimée dans les deux cas. Par contre, le coefficient de portance est très proche de la réalité. Le model Spalart-Allmaras a donné de bons résultats pour $Re=3000000$. Pour le model $k - \varepsilon$, les résultats obtenus diffèrent sensiblement de l'expérimental dans cette plage de Reynolds.

Les différents modèles testés montrent que le logiciel OpenFoam donne de bons résultats dans la modélisation numérique des phénomènes aérodynamiques. Cependant, il est nécessaire d'effectuer certains changements afin de s'adapter au cas étudié ainsi qu'au maillage utilisé. Enfin, le solveur simpleFoam avec le model Spalart-Allmaras semble avoir un bon accord avec la réalité. Il nous semble que les prochaines étapes de validation de ce code pourraient être de tester ce solveur pour un cas pratique.

CHAPITRE 4

CHAPITRE 4

ÉCOULEMENT AUTOUR D'UN AVION SKYWALKER AVEC UN MAILLAGE DYNAMIQUE

4.1. Introduction

Ce chapitre contient les résultats d'une simulation numérique d'un écoulement incompressible autour d'un mini drone radio commandé, plus précisément l'avion Skywalker. C'est aussi un cas pratique de l'analyse aérodynamique d'un problème concret qui a été étudié par nos collègues. Alors que jusqu'à présent uniquement les problèmes de mécanique de fluide bidimensionnel ont été résolus, le chapitre présent comprend un écoulement tridimensionnel autour d'un avion très léger volant à de basses vitesses et à des conditions atmosphériques standards.

L'avion Skywalker est un mini drone radio commandé utilisé principalement pour la cartographie et quelques opérations de la défense militaire. Son rayon d'action dépend essentiellement de ses instruments radio-électroniques et de ses caractéristiques aérodynamiques. C'est pour cela qu'on a décidé de faire une simulation d'un cas pratique afin de découvrir les performances de ce drone.

Le maillage de ce cas va être créé en utilisant un logiciel de conception 3D Solidworks puis exporter sous forme d'un fichier STL (Stéréo lithographie). Afin d'approfondir nos connaissances dans OpenFOAM. On a choisi de faire une simulation 3D avec un maillage dynamique autour de cet avion sans prendre en compte les parties mobiles de l'avion.



Figure 4.1 : Le drone Skywalker.

4.2. Description physique du problème

On considère un avion très léger volant à une vitesse $U = 17\text{m/s}$. Comme le milieu est l'air on a $\nu = 1,46 \times 10^{-5} \text{ m}^2/\text{s}$, et $\rho = 1,225 \text{ kg/m}^3$

Le nombre de Reynolds est égal à

$$\text{Re} = \frac{\rho C|U|}{\mu} = 319720,8$$

Où C est la corde aérodynamique moyenne des ailes est égale à $0,224 \text{ m}$.

Dans le chapitre actuel, il n'y a pas de solution analytique facile à décrire le comportement du fluide. Cependant, il est nécessaire de garder à l'esprit les principales équations qui permettent de résoudre le problème:

L'équation de continuité,

$$\nabla \cdot \mathbf{U} = 0 \quad (4.1)$$

L'équation de quantité de mouvement,

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{U} \quad (4.2)$$

Le nombre de Reynolds,

$$\text{Re} = \frac{C|U|}{\nu} \quad (4.3)$$

Le coefficient de portance,

$$C_z = \frac{F_z}{\frac{1}{2}\rho|U|^2 S} \quad (4.4)$$

Le coefficient de trainée,

$$C_x = \frac{F_x}{\frac{1}{2}\rho|U|^2 S} \quad (4.5)$$

Le coefficient de pression,

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty |U_\infty|^2} \quad (4.6)$$

4.3. Définition d'un maillage dynamique

Le maillage dynamique[13] est une méthode utilisée pour la résolution des problèmes de mécanique des fluides, Il s'agit de la mise au point des schémas en volumes finis pour des maillages généralement non-structurés, mobiles et à topologie éventuellement variable, basés sur des algorithmes intelligents. Ce type de maillage nous permet d'étudier différents écoulements instationnaires où une partie du domaine se déforme se déplace.

4.4. Présentation de l'utilitaire de maillage « snappyHexMesh »

Les problèmes concrets en mécanique de fluide nécessitent souvent un maillage très raffiné dans certaines zones. L'utilitaire snappyHexMesh permet de faire un maillage hexagonal et non structuré avec option de raffinement de couches limites. Pour mailler un domaine autour d'une géométrie quelconque, par exemple un avion, il suffit d'avoir sa géométrie sous format STL qui est obtenue par un logiciel de conception 3D comme Solidworks, de créer un simple maillage de type blockMesh représentant le domaine autour de l'avion et de demander à snappyHexMesh de faire le reste par le fichier system/snappyHexMeshDict.

Pour lancer l'utilitaire snappyHexMesh l'utilisateur doit fournir :

- ✓ Un fichier sous format STL dans le répertoire **constant/triSurface**.
- ✓ Un fichier blockMeshDict qui définit le domaine extérieur.
- ✓ Un fichier snappyHexMeshDict dans le répertoire **system/**, avec les données nécessaires (voir 1.6.1.B).

4.5. Le drone Skywalker

C'est un avion Radio Commander qui est souvent utilisé pour la cartographie et l'exploration des sites isolés. C'est aussi un avion de sport très performant avec son profil Clark-Y et son aile polyédrique.

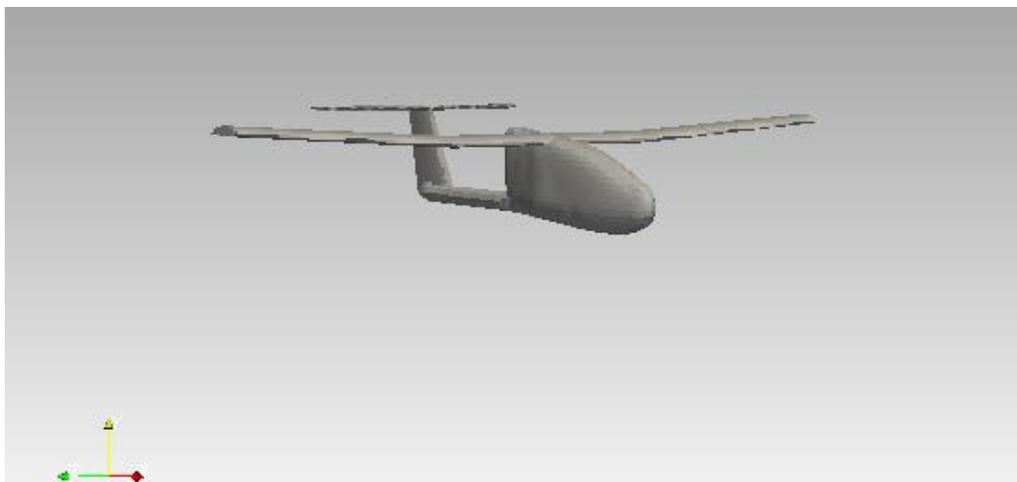


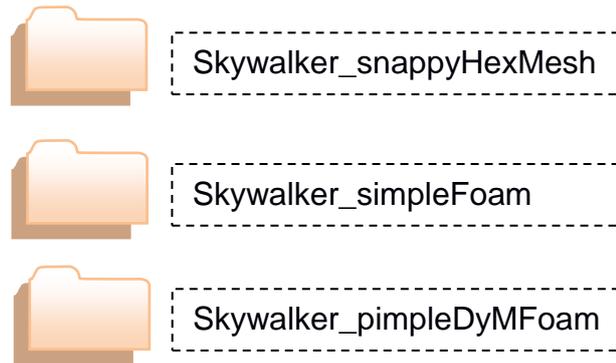
Figure 4.2 : Modélisation de l'avion Skywalker avec Solidworks.

4.6. Résolution numérique

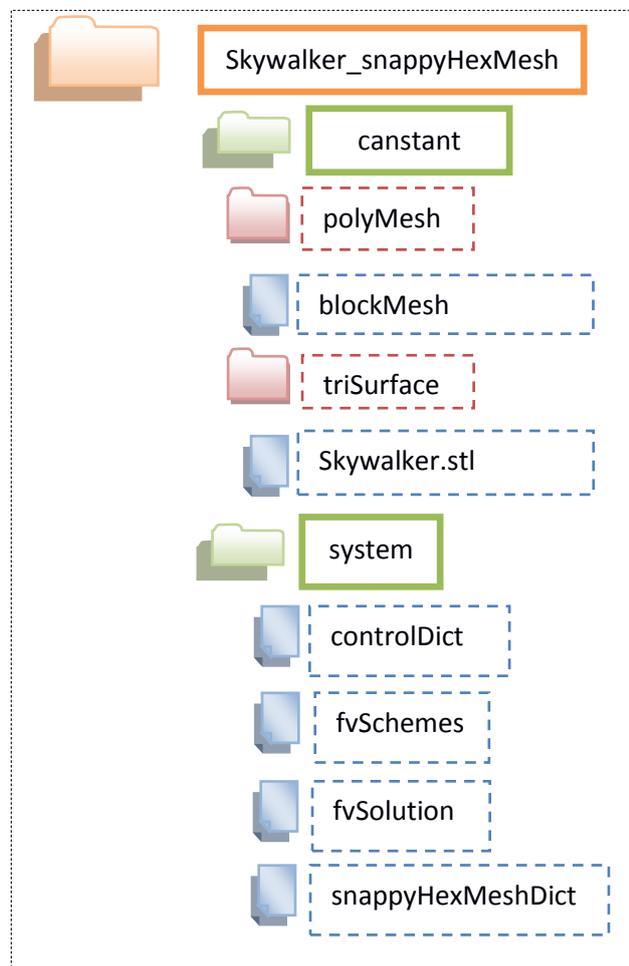
Chaque simulation numérique passe par trois étapes principales: le prétraitement, le traitement et le post-traitement. Comme on la décrit dans le chapitre 1, OpenFOAM a un ensemble de solution pour chacune de ces étapes. Le présent chapitre présente la solution la plus simple pour ce type de problème.

Structure du cas

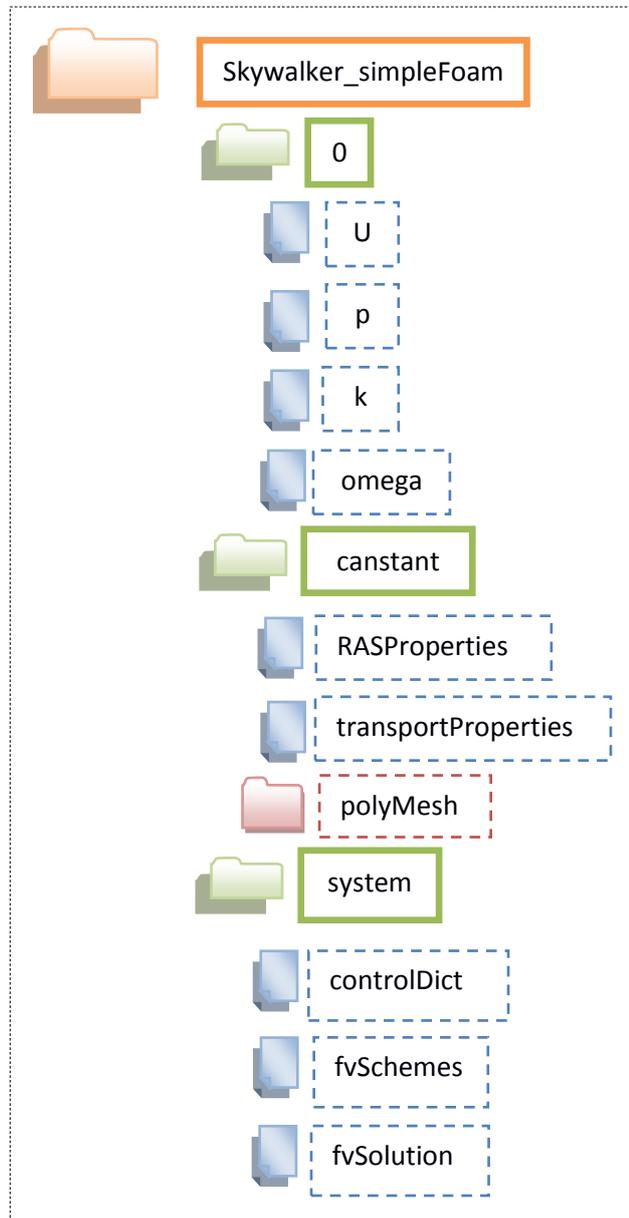
Le dossier qui contient notre cas se compose de trois sous dossiers :



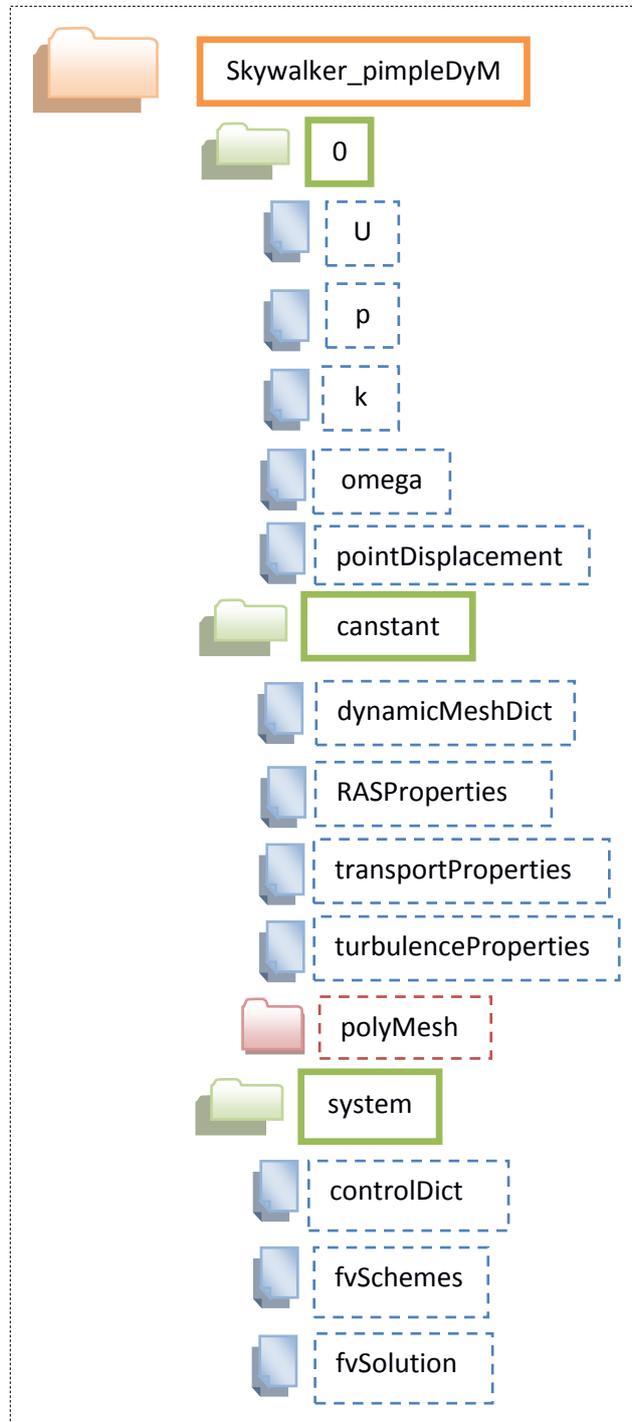
- ✓ Le premier dossier correspond à la phase de prétraitement, c'est le dossier qui permet de faire le maillage du domaine et de l'avion à l'intérieur.



- ✓ Le deuxième dossier correspond à un premier cas de traitement avec un solveur incompressible et stationnaire, simpleFoam, avec une approche de turbulence $k - \omega$ SST.



- ✓ Le troisième dossier présente la dernière phase de traitement et dépôt de traitement avec un solveur incompressible et instationnaire `pimpleDyMFoam` qui contient une option qui permet le déplacement du maillage.



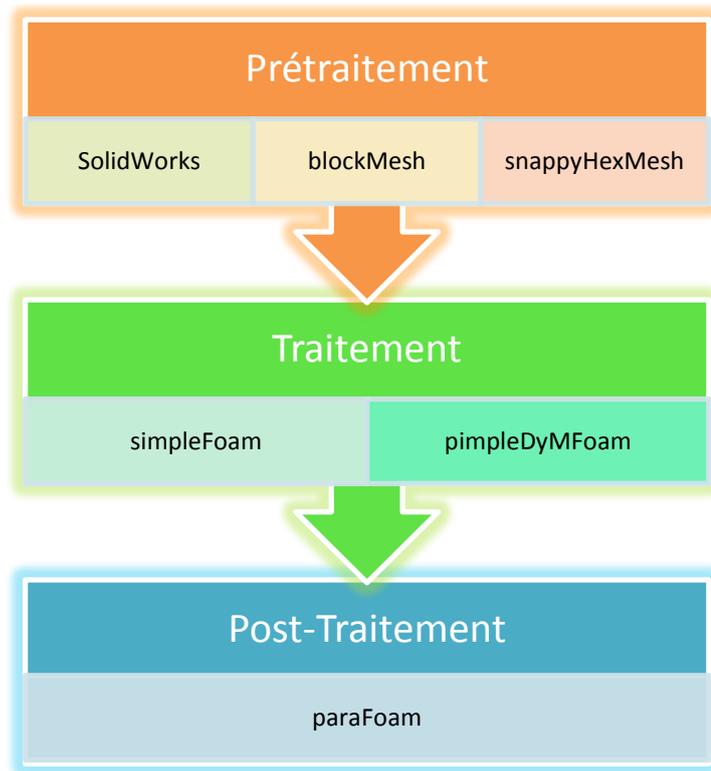


Figure 4.3 : Diagramme des procédures suivies.

4.6.1. Prétraitement

La géométrie de l'avion est générée par un logiciel de conception 3D et importée à OpenFOAM sous forme STL. Le domaine de calcul est ensuite défini sur un fichier blockMeshDict. Dans la dernière étape, il faut donner les instructions nécessaires pour choisir les zones de raffinement et les options du maillage dans un fichier snappyHexMeshDict.

La génération de maillage est initialement faite dans le dossier Skywalker_snappyHexMesh puis exportée vers les autres dossiers pour les calculs. Ce dossier comporte essentiellement deux dictionnaires qui sont blockMeshDict et snappyHexMeshDict.

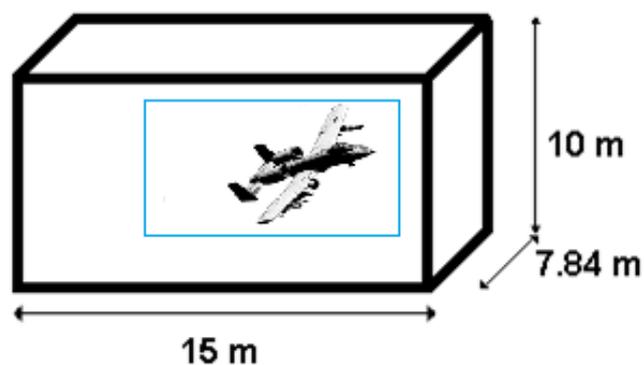


Figure 4.4 : Représentation du domaine de calcul et la zone de raffinement.

Le fichier blockMeshDict :

```

convertToMeters 1;
vertices
(
  (-10 -3.84 -4.16)           //0
  (5 -3.84 -4.16)            //1
  (5 4 -4.16)                //2
  (-10 4 -4.16)             //3
  (-10 -3.84 5.84)          //4   Domaine de calcul
  (5 -3.84 5.84)           //5
  (5 4 5.84)                //6
  (-10 4 5.84)             //7
);
blocks
(
  hex (0 1 2 3 4 5 6 7) (15 6 6) simpleGrading (1 1 1)
);
edges
(
);
boundary // Définition des six faces du domaine
(
  frontAndBack
  {
    type patch;
    faces
    (
      (3 7 6 2)
      (1 5 4 0)
    );
  }
  outlet
  {
    type patch;
    faces
    (
      (0 4 7 3)
    );
  }
  inlet
  {
type patch;
    faces
    (
      (2 6 5 1)
    );
  }
  lowerWall
  {
    type patch;
    faces
    (
      (0 3 2 1)
    );
  }
);

```

```

    }
    upperWall
    {
        type patch;
        faces
        (
            (4 5 6 7)
        );
    }
};

```

Le fichier snappyHexMeshDict :

```

castellatedMesh true;
addLayers true;
geometry
{
    Skywalker.stl // Le fichier contenant la géométrie.
    {
        type triSurfaceMesh;
        name Skywalker;
    }
    refinementBox
    {
        type searchableBox; // Zone de raffinement du maillage.
        min (-6 -1.84 -2.16);
        max ( 2  2 3.84);
    }
};
castellatedMeshControls
{
    maxLocalCells 100000; // Paramètres de raffinement du maillage
    maxGlobalCells 2000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 3;
    refinementSurfaces // Surfaces de raffinement
    {
        Skywalker
        {
            level (5 6);
            patchInfo // Informations sur les faces de raffinement
            {
                type wall; // type des faces
                inGroups (SkywalkerGroup); // Noms des faces
            }
        }
    }
    locationInMesh (3.0001 3.0001 0.43); // Position dans le domaine
    allowFreeStandingZoneFaces true;
}
addLayersControls // Paramètres de la couche limite
{
    relativeSizes true;
}

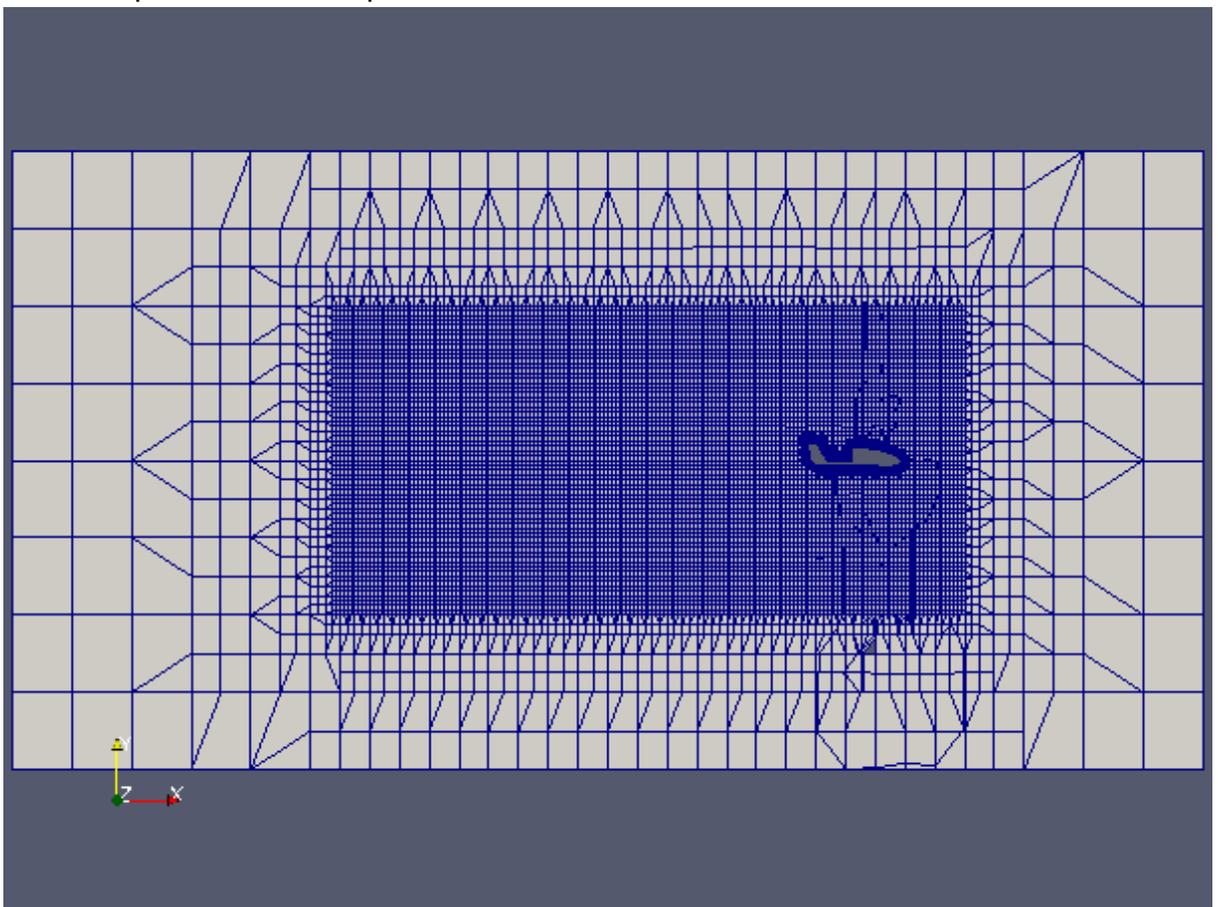
```

```

layers
{
  "(Skywalker).*" // Liste des faces avec une couche limite
  {
    nSurfaceLayers 1;
  }
} // Paramètres de raffinement de la couche limite
expansionRatio 1.0;
finalLayerThickness 0.3;
minThickness 0.1;
nGrow 0;
featureAngle 60;
slipFeatureAngle 30;
nRelaxIter 3;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedianAxisAngle 90;
nBufferCellsNoExtrude 0;
nLayerIter 50;
}

```

Le maillage obtenu est non structuré. Il se compose principalement des mailles hexaédriques et tétraédriques avec raffinement de la couche limite.



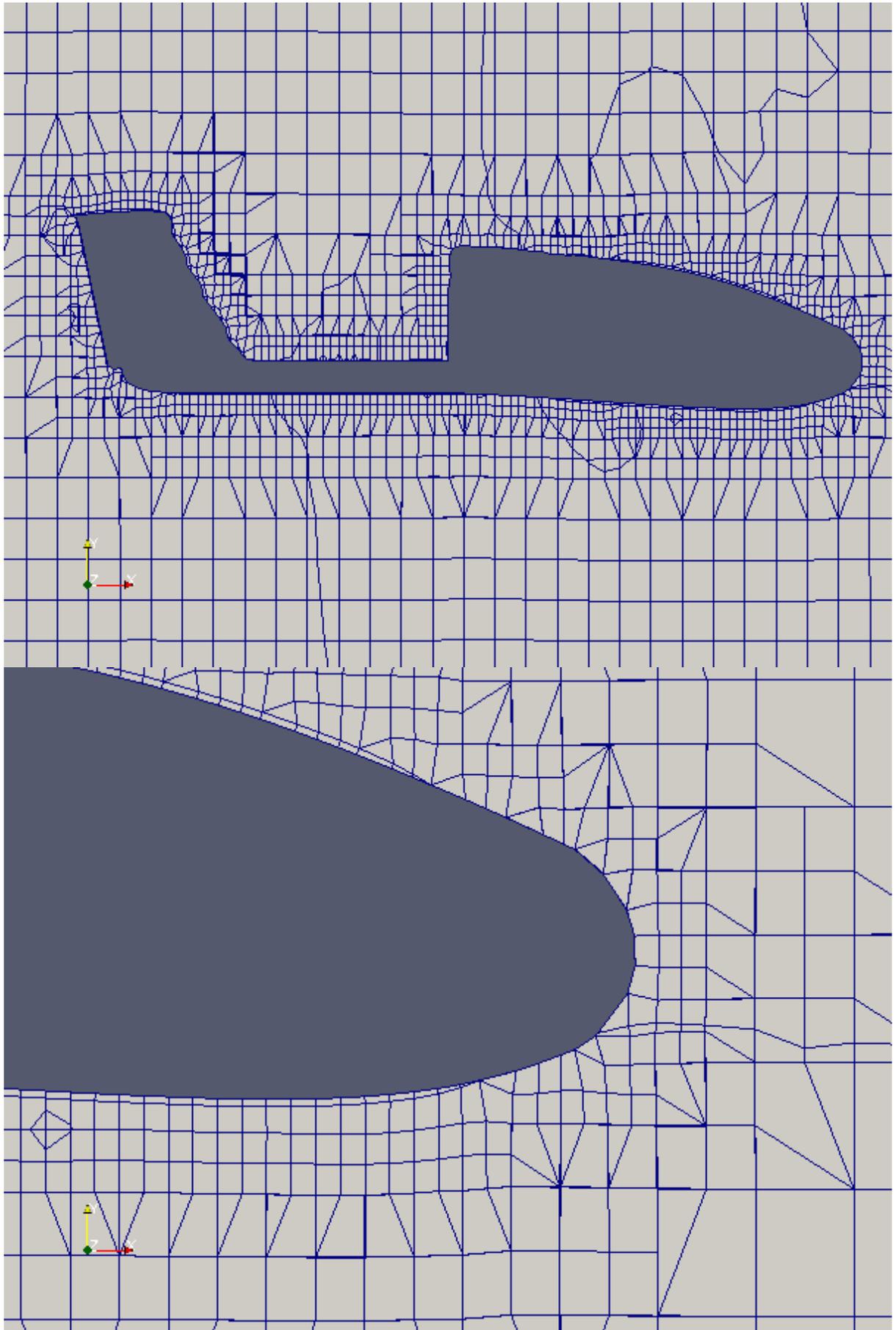


Figure 4.5 : Visualisation du maillage obtenu par snappyHexMesh.

4.6.2. Traitement

Dans cette partie on a choisi de diviser le travail en deux parties, la première consiste à importer le maillage obtenu par snappyHexMesh et de lancer les calculs pour un angle de 0° par le solveur stationnaire simpleFoam. Ensuite, la deuxième concerne l'importation des résultats de la première partie et le lancement des calculs avec le solveur pimpleDyMFoam qui est élaboré pour les calculs dans les cas instationnaires avec déplacement du maillage suivant un dictionnaire nommé dynamicMeshDict.

4.6.2.1. Choix du solveur pimpleDyMFoam

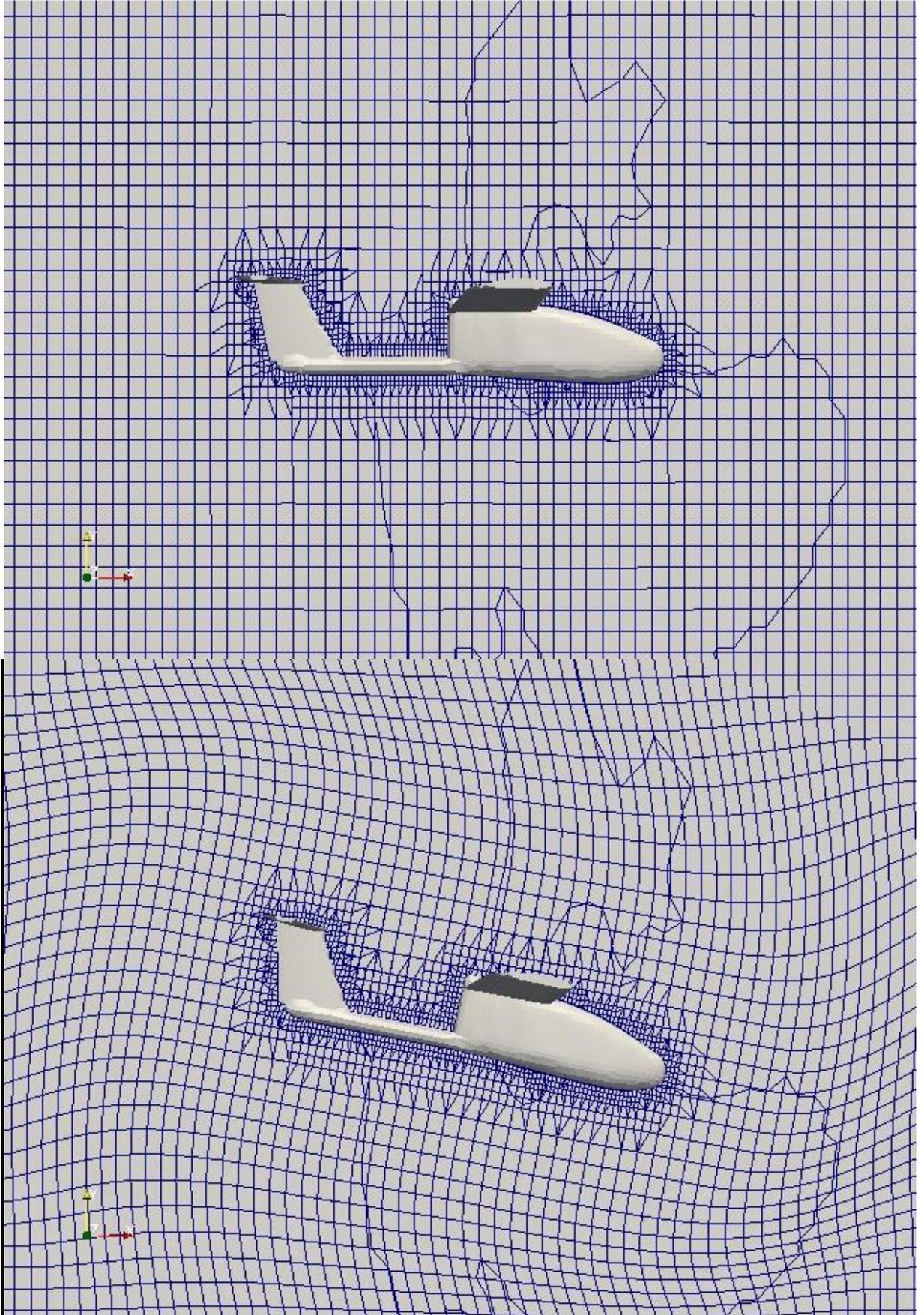
Le solveur pimpleDyMFoam est destiné à la simulation des écoulements incompressibles instationnaires avec l'implémentation du maillage dynamique.

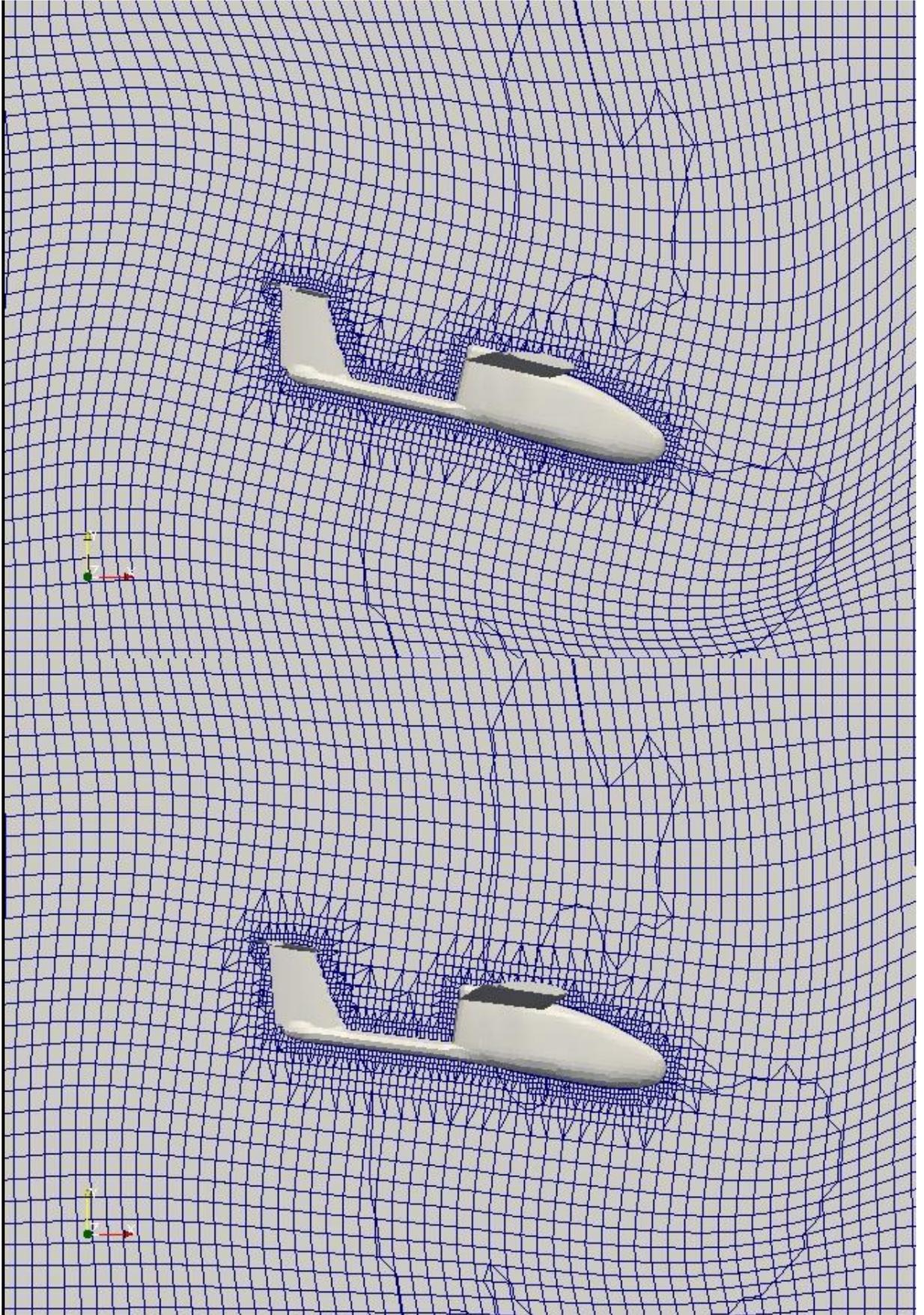
Le fichier Skywalker_pimpleDyMFoam/0/pointDisplacement correspond aux déplacements des frontières du domaine.

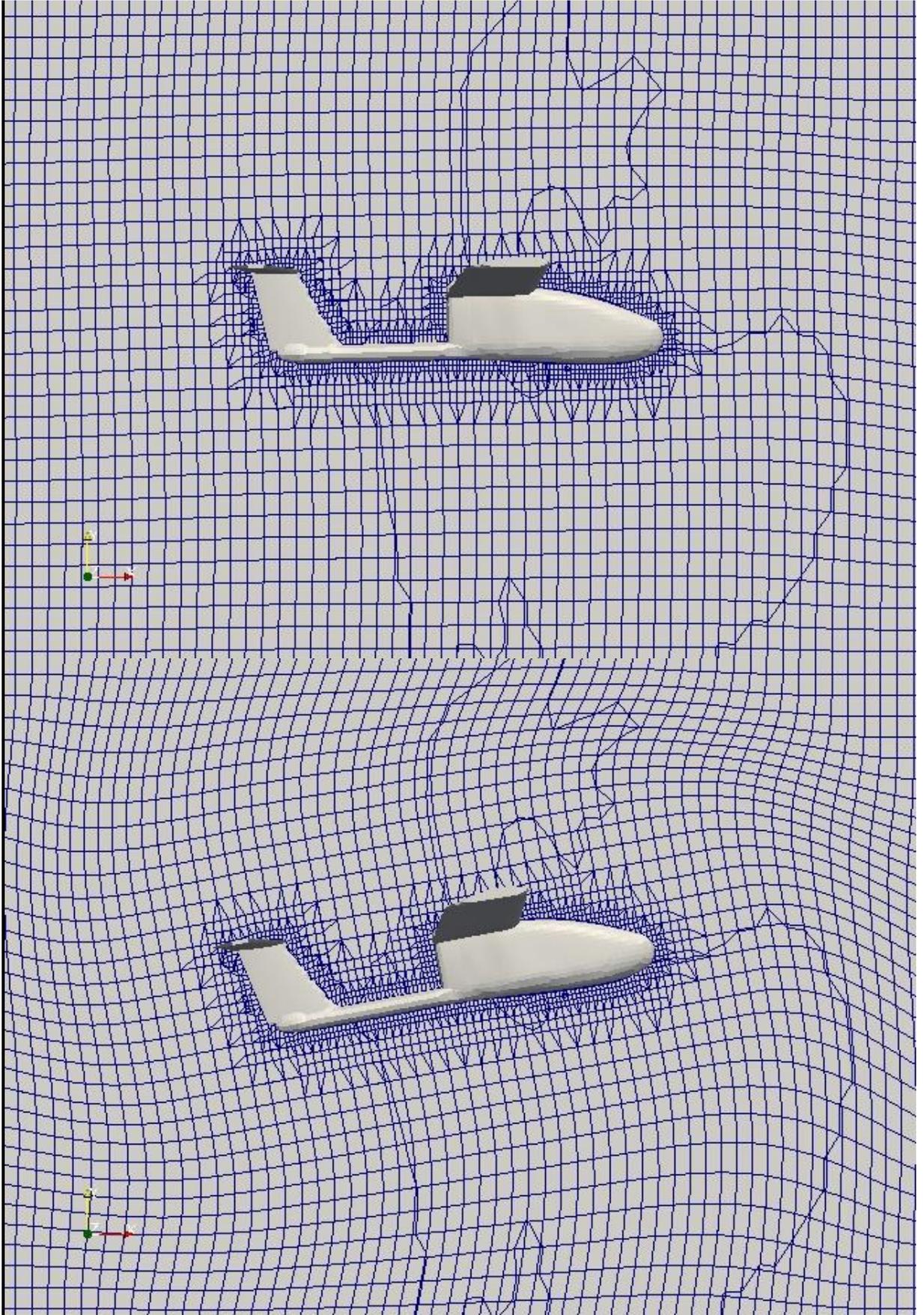
```

dimensions [0 1 0 0 0 0]; // Dimensions en mètre
internalField uniform (0 0 0); // Déplacements initiales des maille à l'intérieur.
boundaryField
{
    Skywalker
    {
        type calculated; // Déplacements du drone à l'intérieur est calculé.
        value uniform (0 0 0);
    }
    "*"
    {
        type fixedValue; // Tous les mailles sur les frontières du domaine
        value uniform (0 0 0); // restent fixes.
    }
}

```







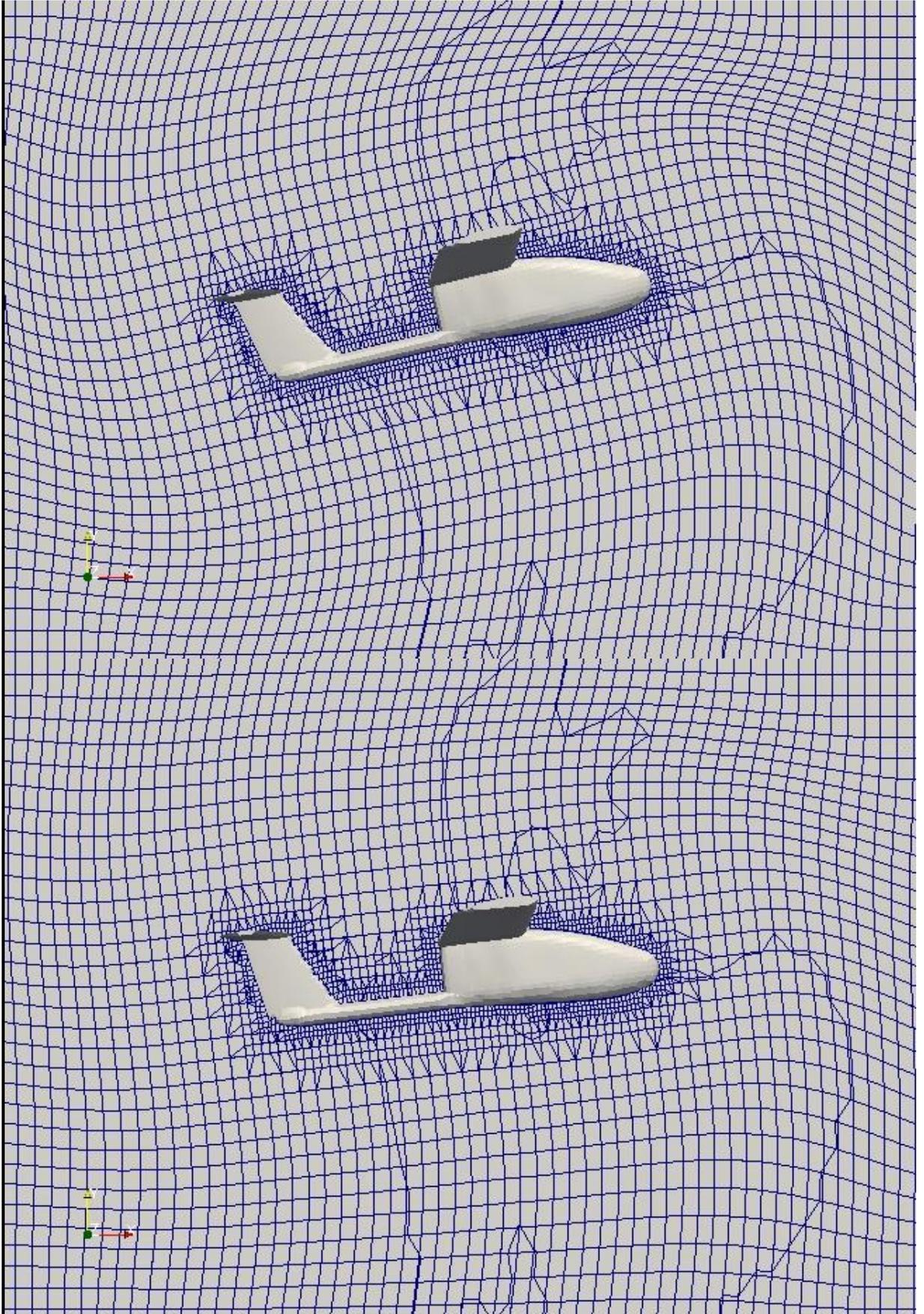


Figure 4.6 : Visualisation du maillage dynamique.

On remarque que les mailles sont en mouvement par rapport à l'avion tout en restant parallèle à l'écoulement ce qui nous donne un précieux avantage dans les calculs.

4.6.2.2. Conditions aux limites

Tableau 4.1 : Conditions aux limites.

Paramètres	Faces	Conditions aux limites	Description
U	Inlet	fixedValue, uniform (17 0 0)	Vitesse uniforme 17m/s.
	Outlet	inletOutlet	Ecoulement libre dans la sortie.
	SkywalkerGroup	movingWallVelocity, uniform (0 0 0)	Conditions de non glissement.
	frontAndBack	slip	Condition de glissement de la paroi.
	lowerWall	slip	//
	upperWall	slip	//
p	Inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	Outlet	fixedValue, uniform 0	Pression fixe (nulle).
	SkywalkerGroup	zeroGradient	Gradient nulle dans la direction de la normale.
	frontAndBack	slip	Condition de glissement de la paroi.
	lowerWall	slip	//
	upperWall	slip	//
K	Inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	Outlet	fixedValue, uniform 0	Pression fixe (nulle).
	SkywalkerGroup	zeroGradient	Gradient nulle dans la direction de la normale.
	frontAndBack	slip	Condition de glissement de la paroi.
	lowerWall	slip	//
	upperWall	slip	//
omega	Inlet	zeroGradient	Gradient nulle dans la direction de la normale.
	Outlet	fixedValue, uniform 0	Pression fixe (nulle).
	SkywalkerGroup	zeroGradient	Gradient nulle dans la direction de la normale.
	frontAndBack	slip	Condition de glissement de la paroi.
	lowerWall	slip	//
	upperWall	slip	//

4.6.3. Post-traitement

Dans cette partie les résultats obtenus sont automatiquement adaptés pour être visualisé sous Paraview, le logiciel de post-traitement standard d'OpenFOAM. Le calcul des coefficients de portance et de trainé sont obtenus en introduisant une fonction dans le fichier « controlDict » pareille au cas précédant. La visualisation des lignes de courants est obtenue grâce à la fonction « Stream Lines » de Paraview.

Les résultats obtenus sont pour différents angles d'attaque à différents instants. On a également introduit une fonction pour calculer les coefficients de portance et de trainée pour cet avion afin de les comparer avec ceux obtenus par nos camarades: $C_x = 0.0013 + 0.49C_z^2$. On peut remarquer une différence entre les deux courbes qui est due surtout au fait que notre calcul est pour le cas instationnaires et notre drone est continuellement en mouvement.

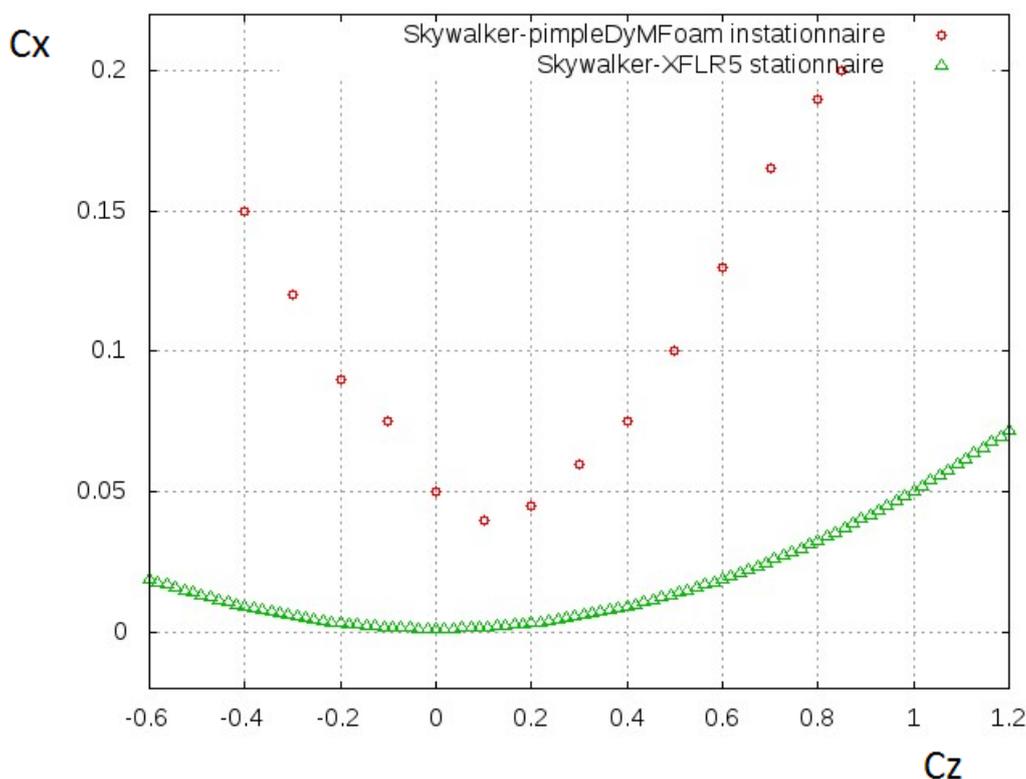
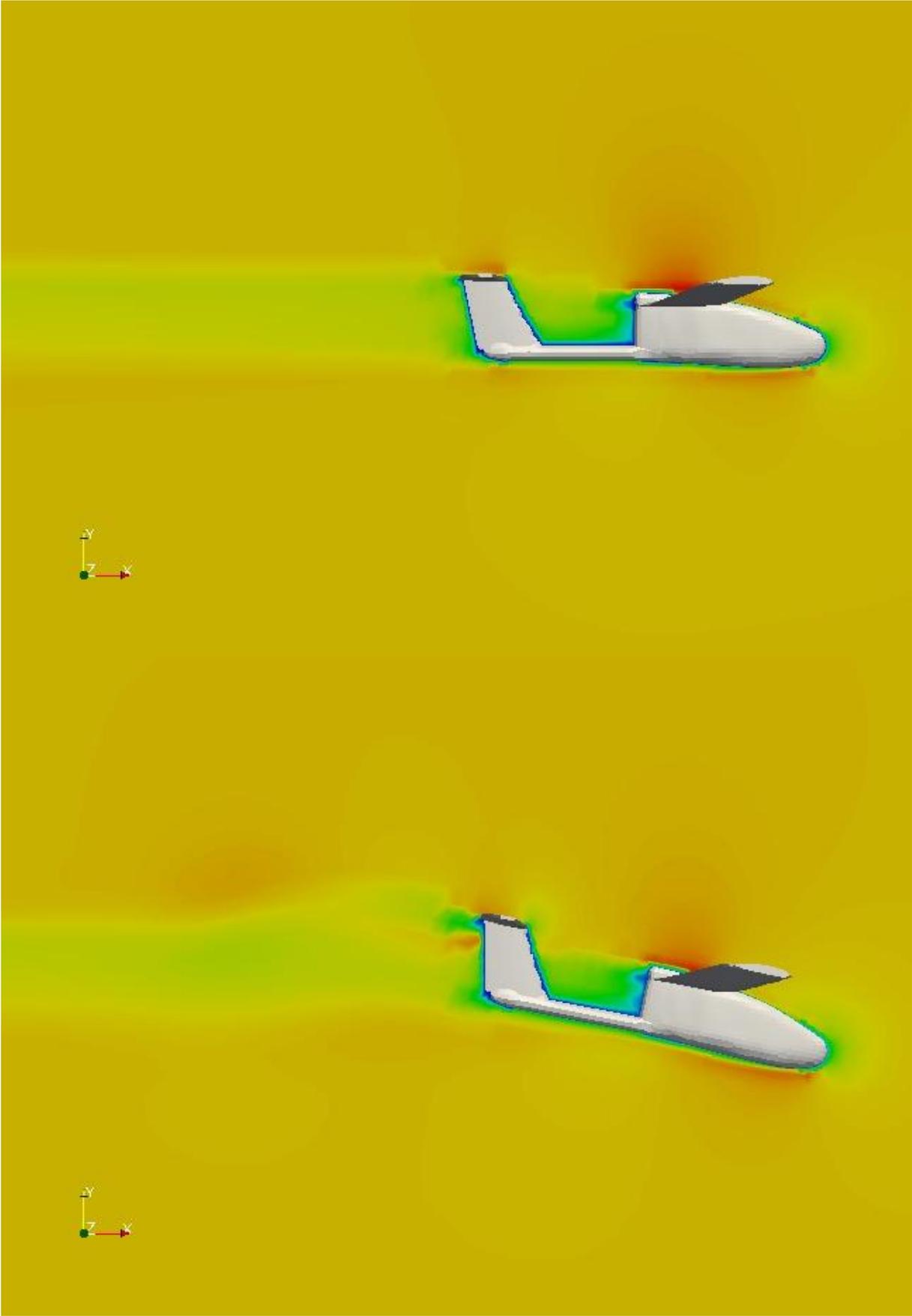
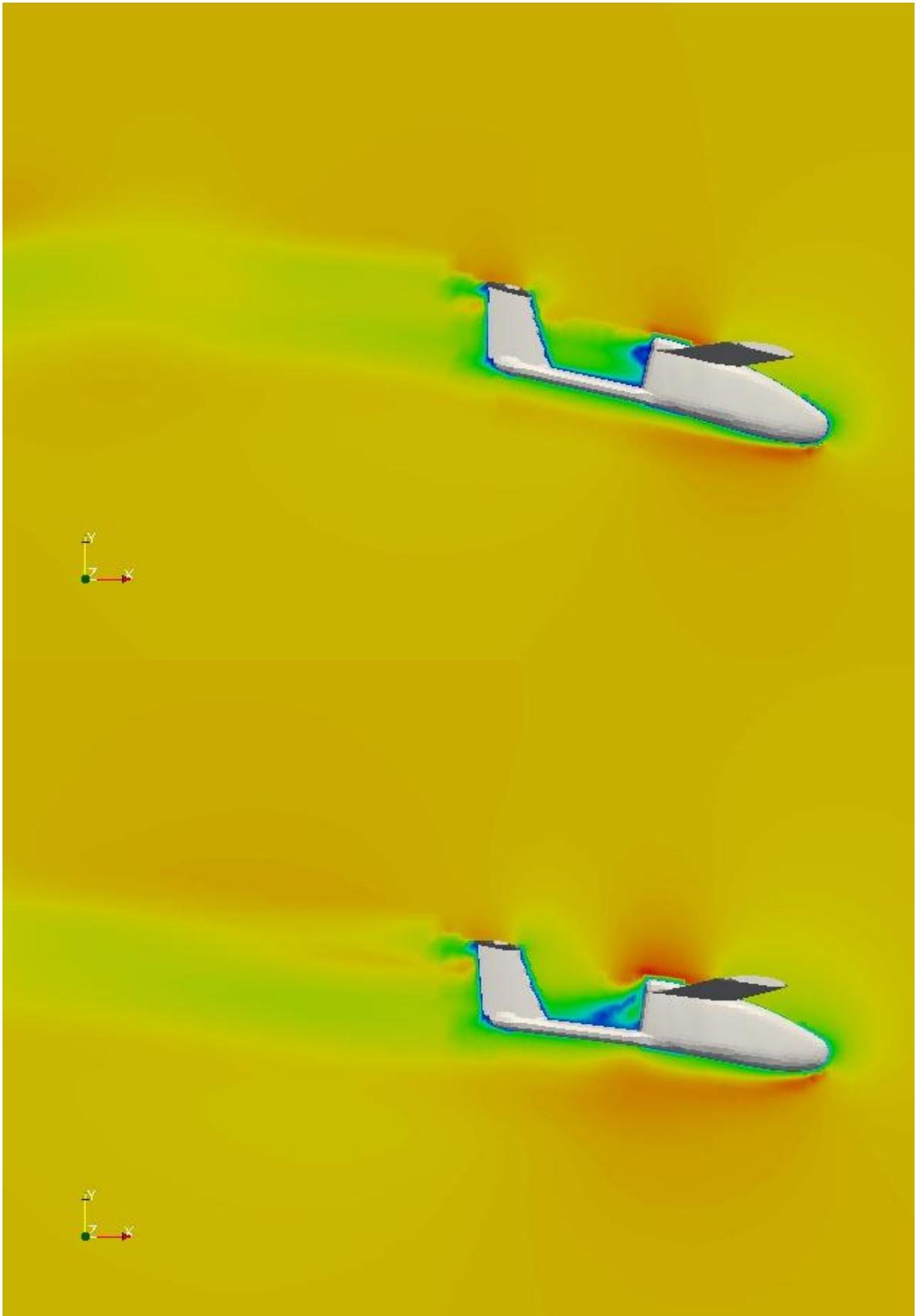


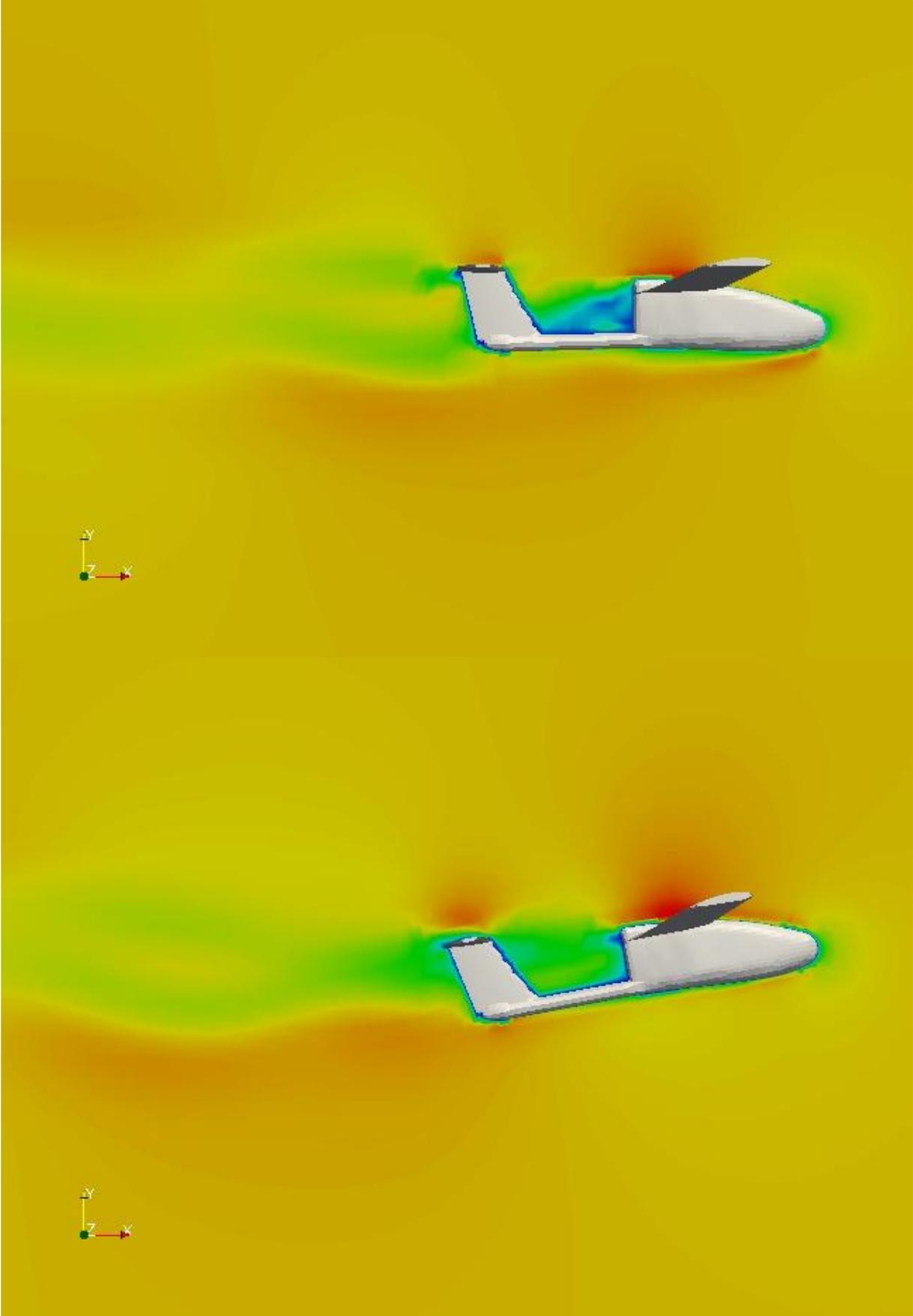
Figure 4.7 : Comparaison de la polaire du drone Skywalker.

- Champs de vitesse :

Le champ de vitesse est visualisé sur Paraview comme un plan vertical qui passe par le centre de l'avion.







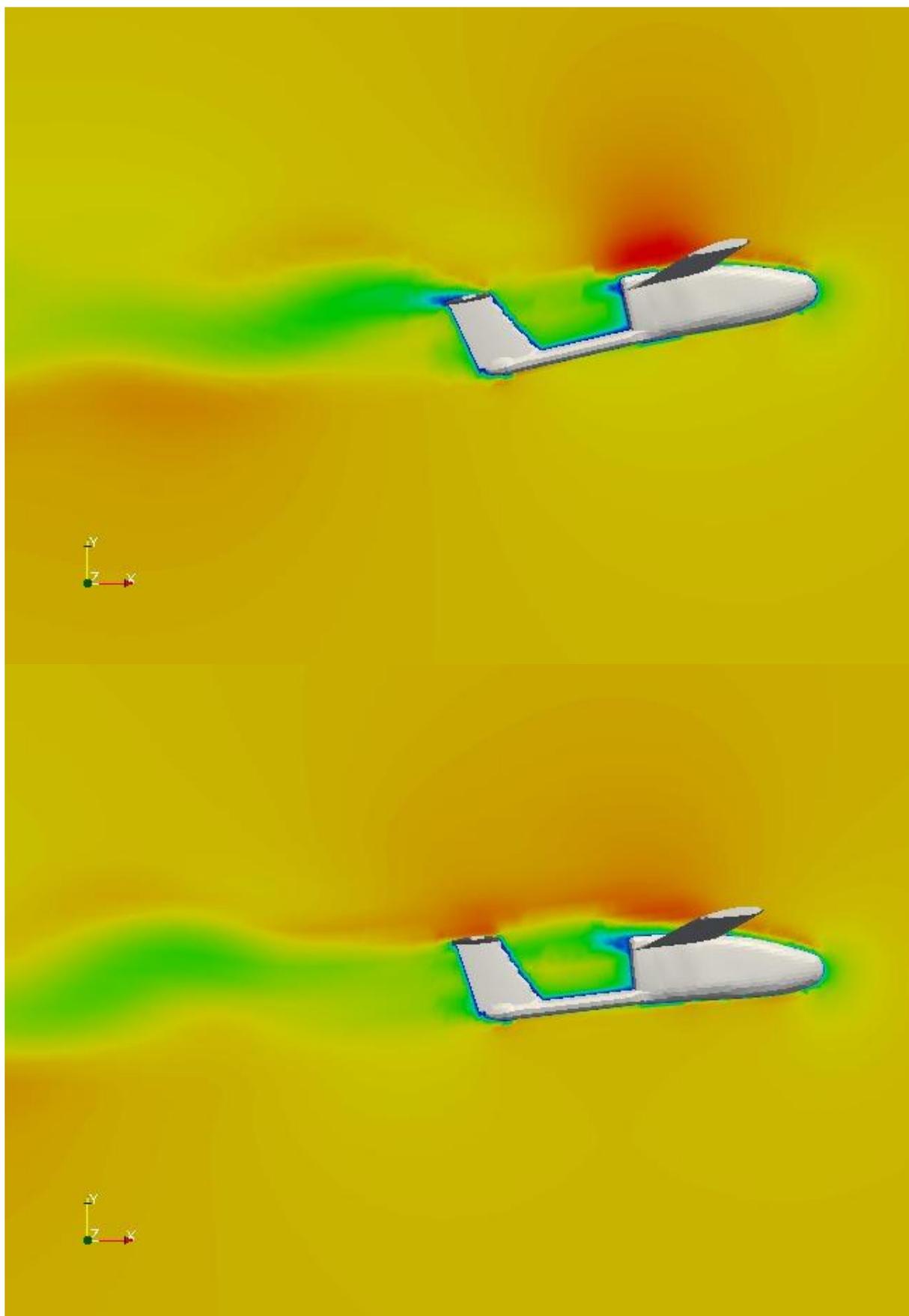
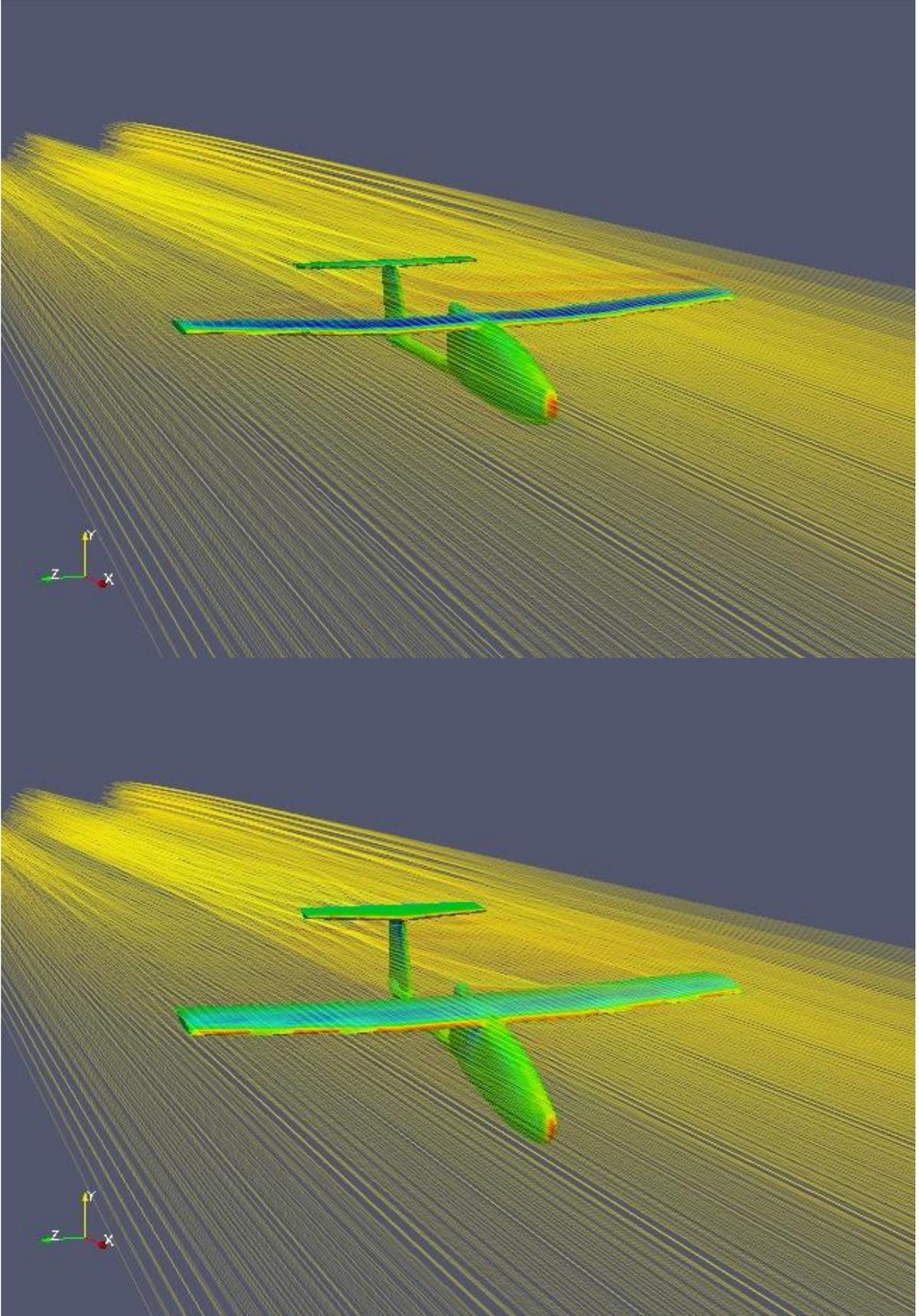
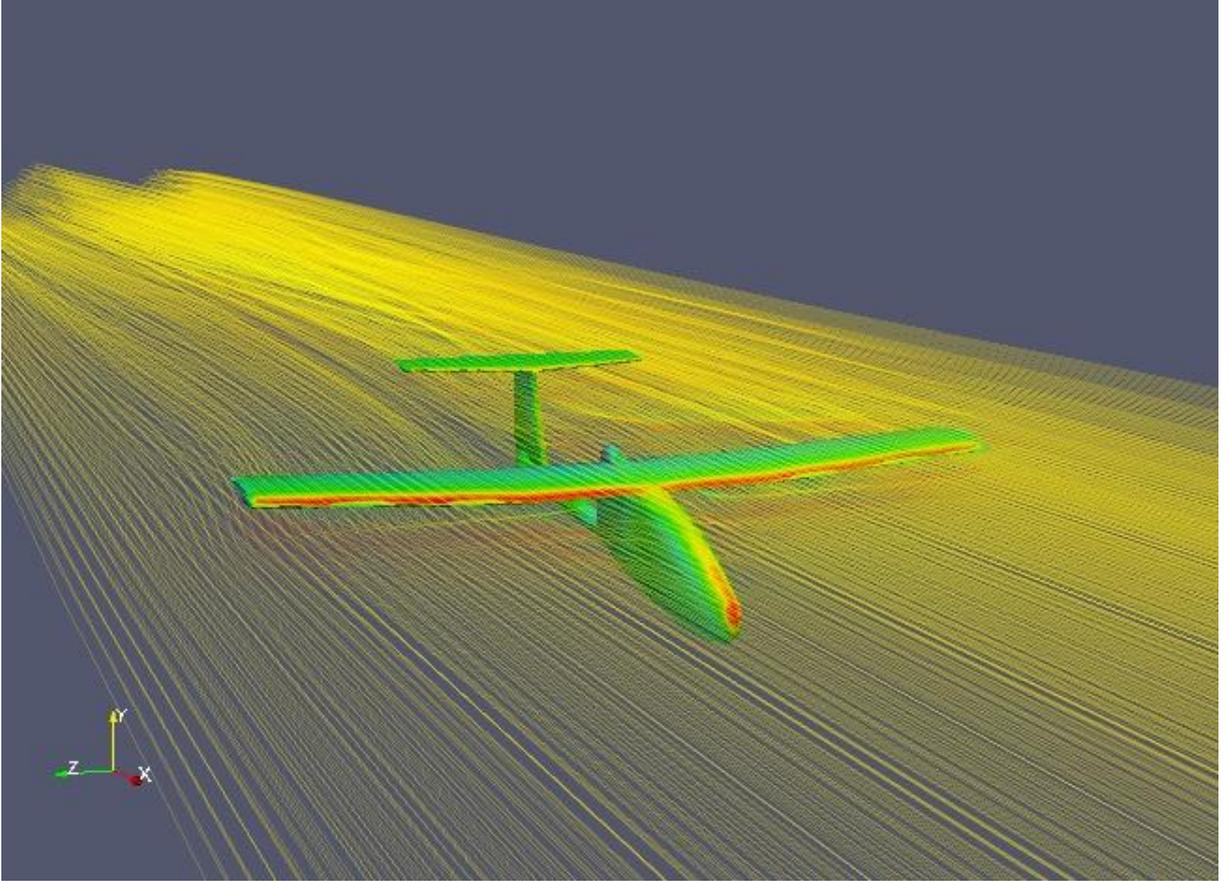
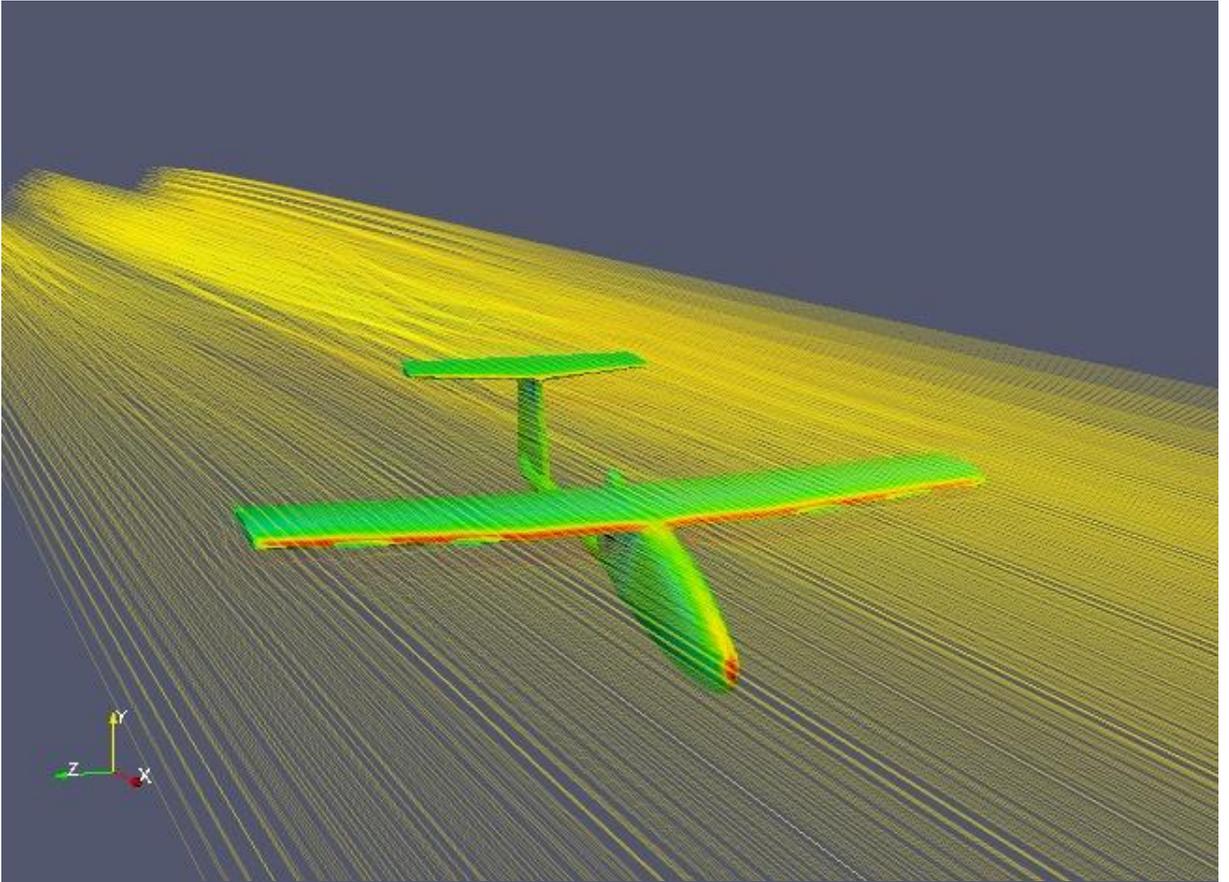


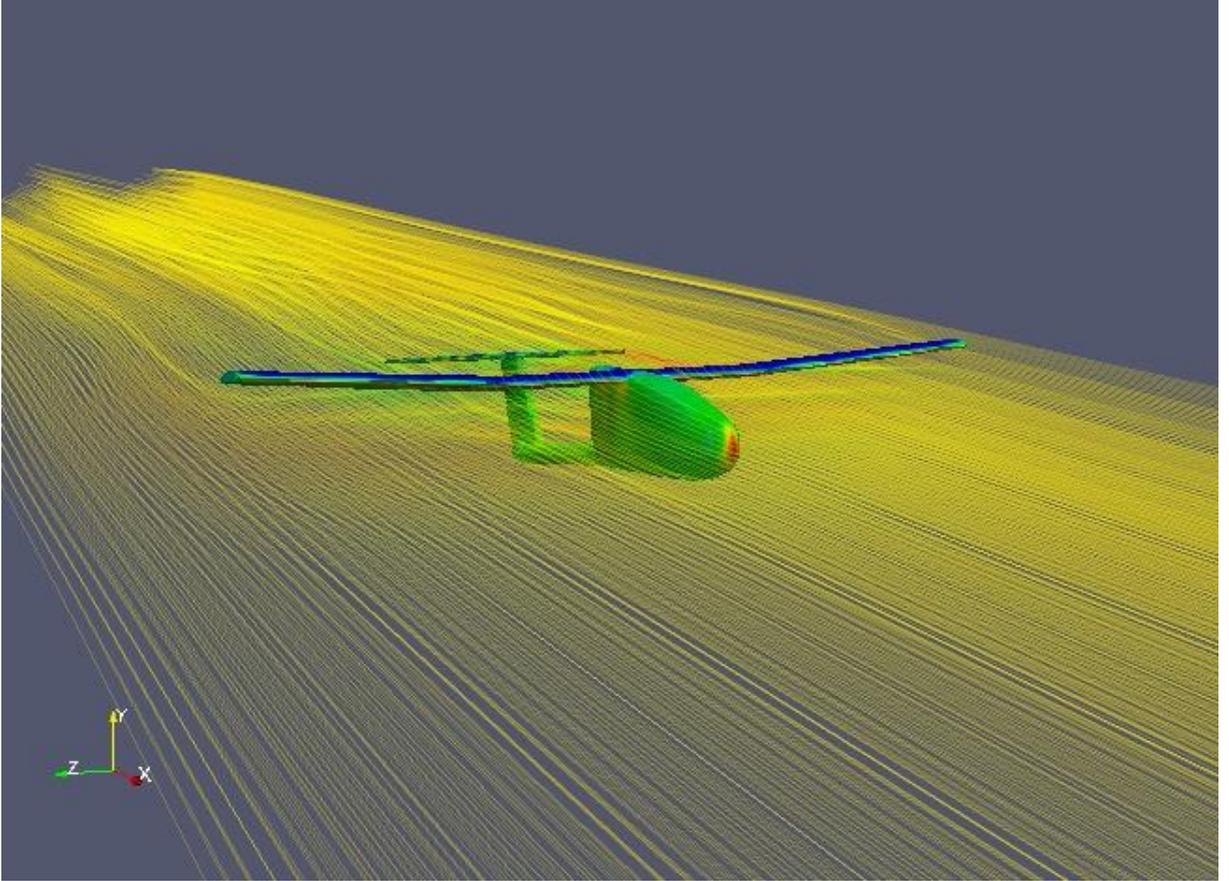
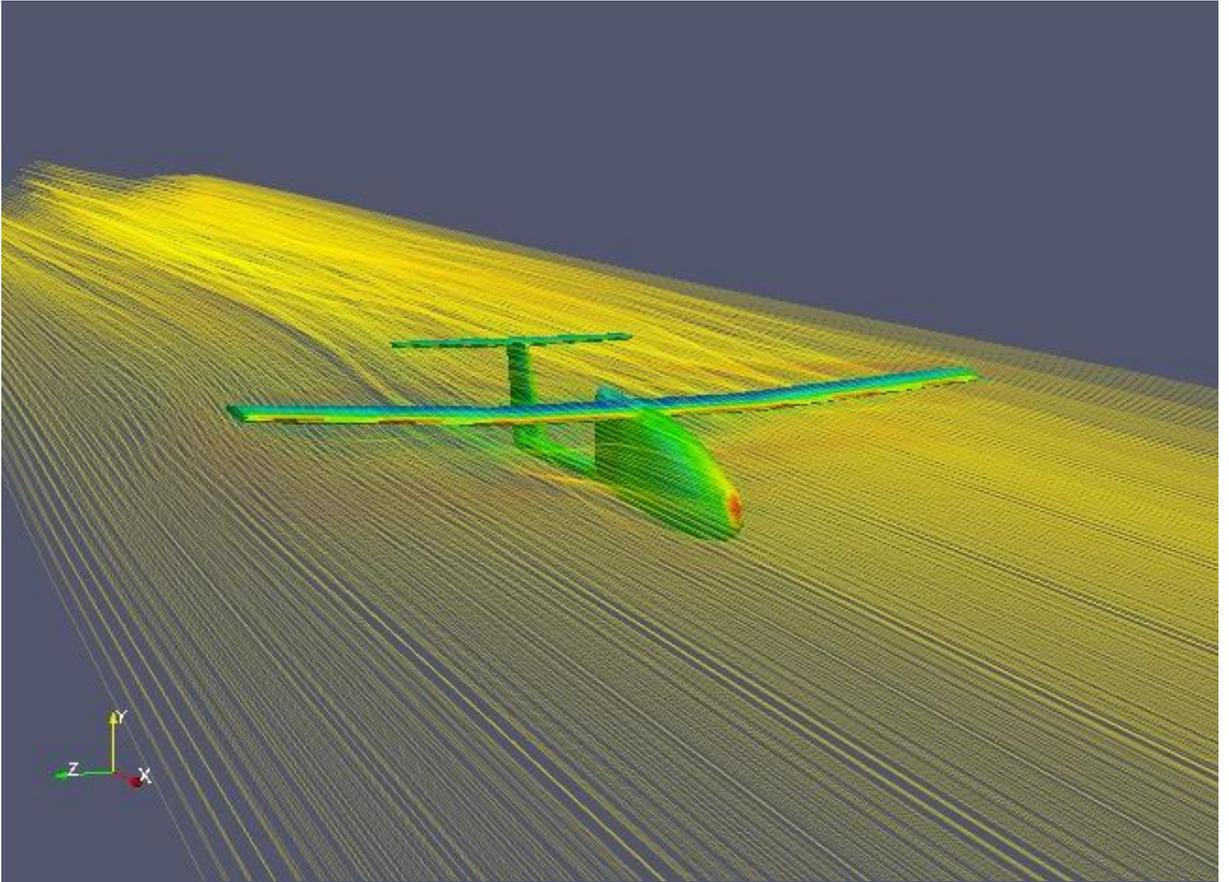
Figure 4.8 : Visualisation du champ de vitesse

- Lignes de courants :

Ce sont les lignes de trajectoire des particules qui se déversent autour de l'avion.
[A3].







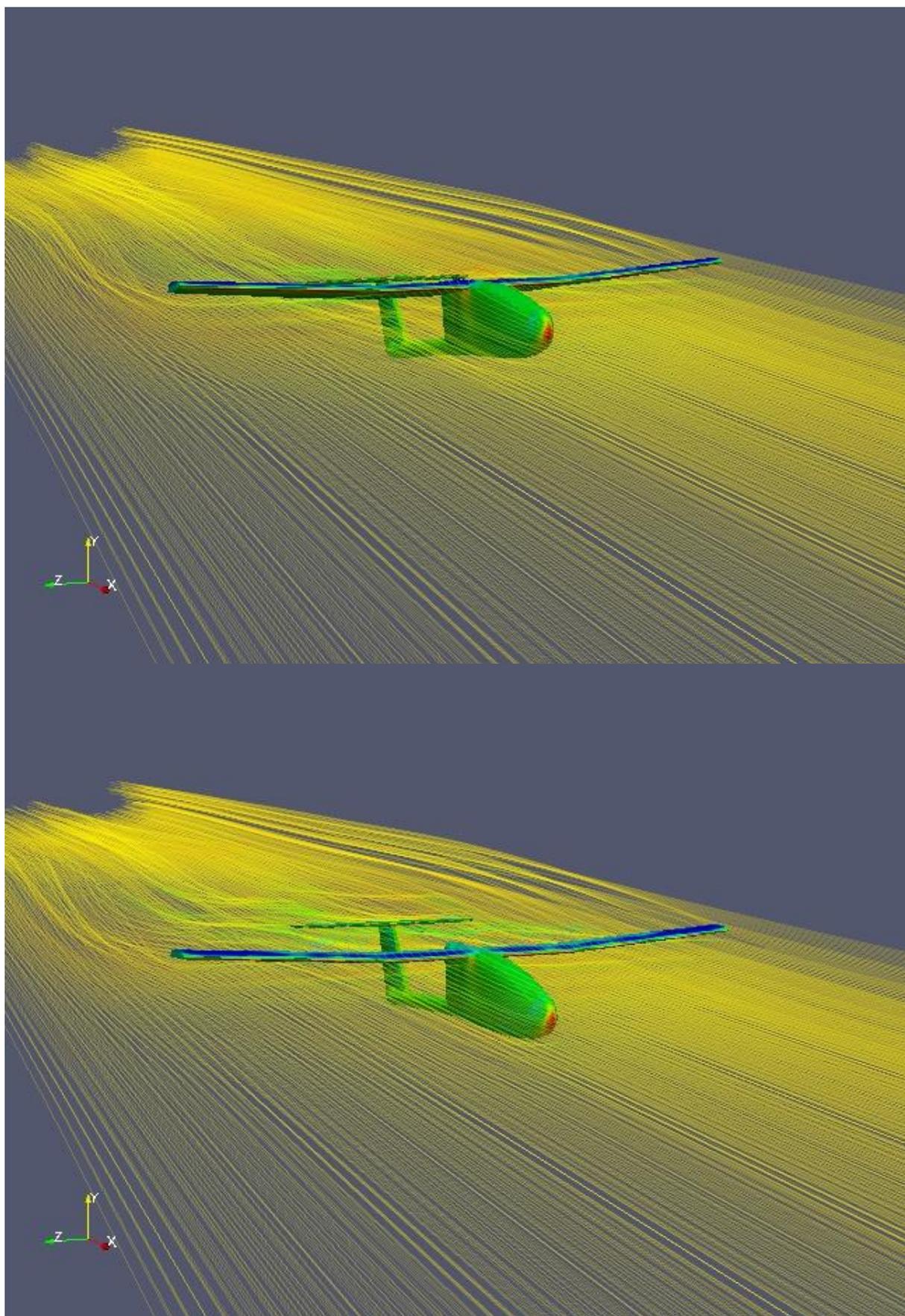


Figure 4.9 : Visualisation des lignes de courants autour de l'avion.

4.7. Conclusion

On peut conclure de ce chapitre qu'OpenFOAM est un solveur très puissant qui permet la simulation des écoulements autour des objets en mouvement. On peut aussi prouver que le maillage du domaine, dans un cas d'écoulement externe autour d'un corps donné est très simple avec l'utilitaire de maillage snappyHexMesh d'OpenFOAM. Les calculs instationnaires sont plus proches de la réalité mais ils prennent beaucoup plus de temps.

Cette étude peut aller plus loin si on prend en considération les parties mobiles de l'avion (les ailerons, les gouvernes, ...) afin d'entamer un calcul approfondi de stabilité de cet aéronef.

CHAPITRE 5

CHAPITRE 5

ÉCOULEMENT SUPERSONIQUE DANS UN DIVERGEANT

5.1. Introduction

Les études expérimentales sur les écoulements compressibles supersoniques dans les tuyères convergentes-divergentes, ont prouvées l'existence et l'interaction de plusieurs phénomènes physiques : jet supersonique, ondes de choc, décollement de la couche limite, écoulement de retour, couche de mélange visqueuse et turbulente. Ces phénomènes très complexes peuvent considérablement influencer sur les performances et la fiabilité des tuyères propulsives.

Dans le présent chapitre on simule un divergeant d'une tuyère supersonique traversée par un fluide considéré comme gaz parfait. On s'intéresse ensuite sur l'effet de la forme de section sur la nature de l'écoulement à l'intérieur de la tuyère. La difficulté dans ce problème réside dans l'importation de la géométrie des différents types de divergeant.



Figure 5.1 : Différentes géométries de tuyère supersonique.

5.2. Description physique

Une tuyère supersonique [14] est une conduite de forme convergente-divergente qui a un profil spécifiquement étudié. Elle met en communication une chambre contenant un gaz comprimé avec un milieu extérieur d'évacuation. La chambre et le milieu extérieur sont à des pressions différentes pour permettre le mouvement du gaz. La tuyère supersonique a pour rôle d'accélérer les gaz d'une vitesse faible à l'entrée du convergent jusqu'à une vitesse sonique au col puis à une vitesse supersonique à la sortie du divergeant de la tuyère.

La forme de la tuyère permet de détendre les gaz. L'opération de détente transforme l'énergie potentielle du gaz en énergie cinétique. Tout le long de la tuyère, la vitesse du gaz augmente, alors que sa pression et sa température diminuent. La tuyère supersonique est appelée tuyère de Laval, elle est représentée dans la figure 5.2.

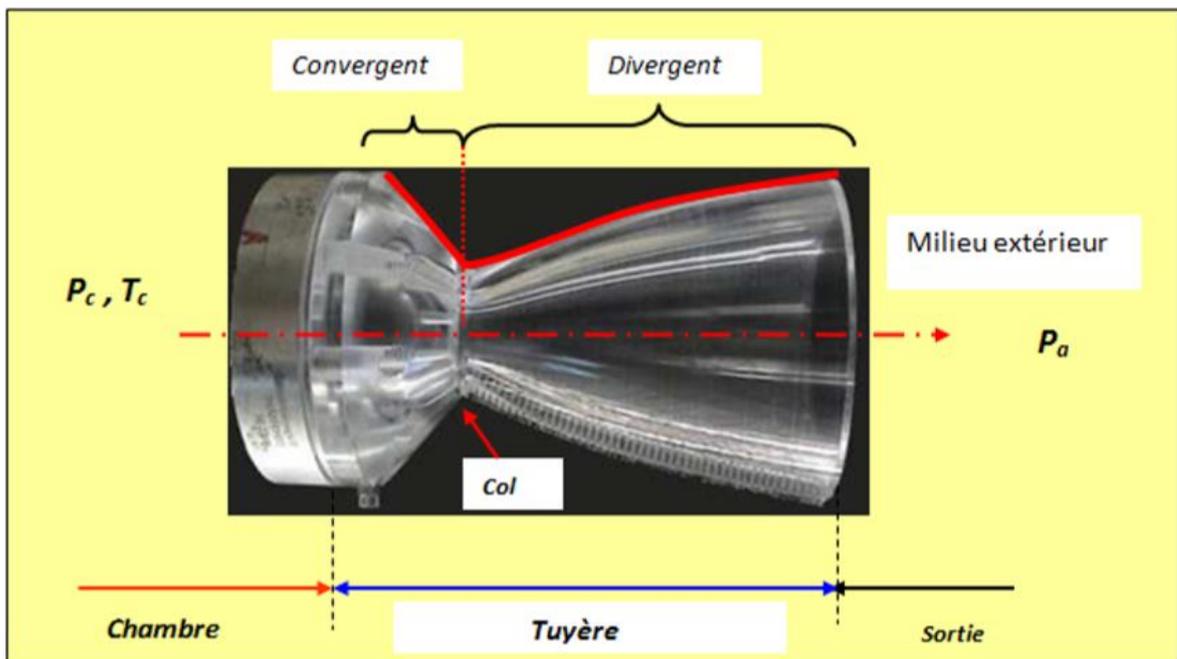


Figure 5.2 : La tuyère supersonique. [15]

L'approche monodimensionnelle des écoulements des gaz calorifiquement parfait et non-visqueux permet de décrire les différents régimes de fonctionnement d'une tuyère supersonique.

Considérons le système représenté par la figure 5.3, constitué d'une tuyère Laval. La tuyère est en communication avec un réservoir contenant un gaz à une pression génératrice P_c supposée fixe. Le divergeant débouche dans une enceinte quasi infinie dont la pression statique P_a est variable.

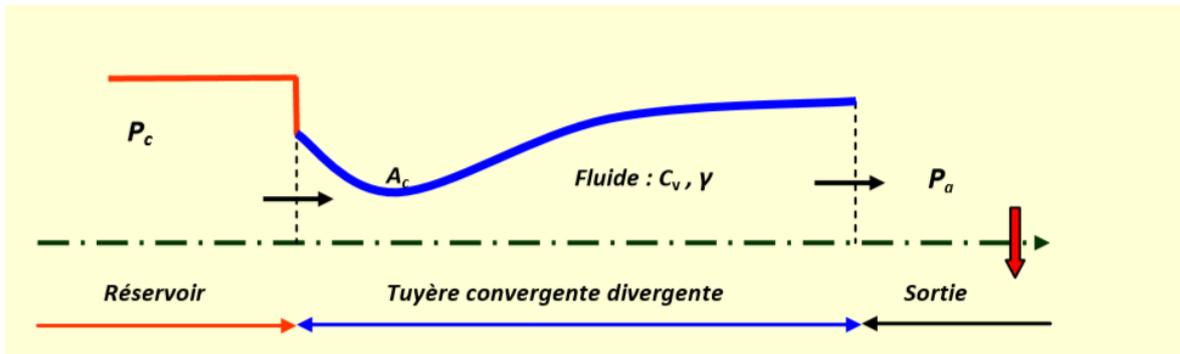


Figure 5.3 : Schéma de la tuyère supersonique. [15]

Si la pression du milieu extérieur est égale à la pression de la chambre, le fluide est entièrement au repos (figure 5.5, courbe (a)). Si la pression du milieu ambiant P_a diminue progressivement au-dessous de la pression P_c supposée constante, un débit d'écoulement prend naissance et cinq régimes particuliers d'écoulements naissent.

5.2.1. Ecoulement subsonique

Au début de l'écoulement, le débit augmente avec la diminution de la pression ambiante. L'écoulement est subsonique dans toute la tuyère [figure 5.5, courbes (b) et (c)]. Dans le convergent la vitesse augmente et la pression diminue jusqu'à une valeur minimale atteinte au col. Dans le divergent la vitesse diminue et la pression augmente jusqu'à la pression du milieu extérieur. La pression est donc minimale dans la section du col où la vitesse passe par une valeur maximale.

5.2.2. Ecoulement sonique adapté

Si P_a continue à diminuer jusqu'à une valeur caractéristique notée P_{ac1} , l'écoulement au col devient sonique et le débit atteint une valeur limite maximale [figure 5.5, courbe (d)]. L'écoulement est dit sonique adapté. Partout encore, sauf au col, l'écoulement est subsonique. Au col de la tuyère, la pression atteint la valeur critique P^* et la vitesse atteint celle du son. .

5.2.3. Ecoulement sonique non adapté

Si P_a diminue au-dessous de P_{ac1} , le col reste toujours sonique. Dans le divergent apparaît un phénomène irréversible appelé onde de choc, l'écoulement est dit sonique non adapté ou régime de sur-détente. L'évolution de l'écoulement du gaz est représentée par la courbe (e) de la figure 5.5. Dans le convergent l'écoulement ne subit aucune modification mais dans le divergent une onde de choc stationnaire et normale à l'axe de la tuyère divise l'écoulement en deux parties (figure 5.4). Dans la première partie du divergent l'écoulement est supersonique, la vitesse continue à augmenter et la pression diminue. A travers l'onde de choc, la pression augmente brusquement, et la vitesse diminue brusquement. L'onde de choc se place de telle façon que la recompression qui s'effectue ramène la pression de la sortie à la pression ambiante P_a . Dans la deuxième partie du divergent (après l'onde de choc) la pression remonte et la vitesse diminue progressivement, l'écoulement est à nouveau subsonique.

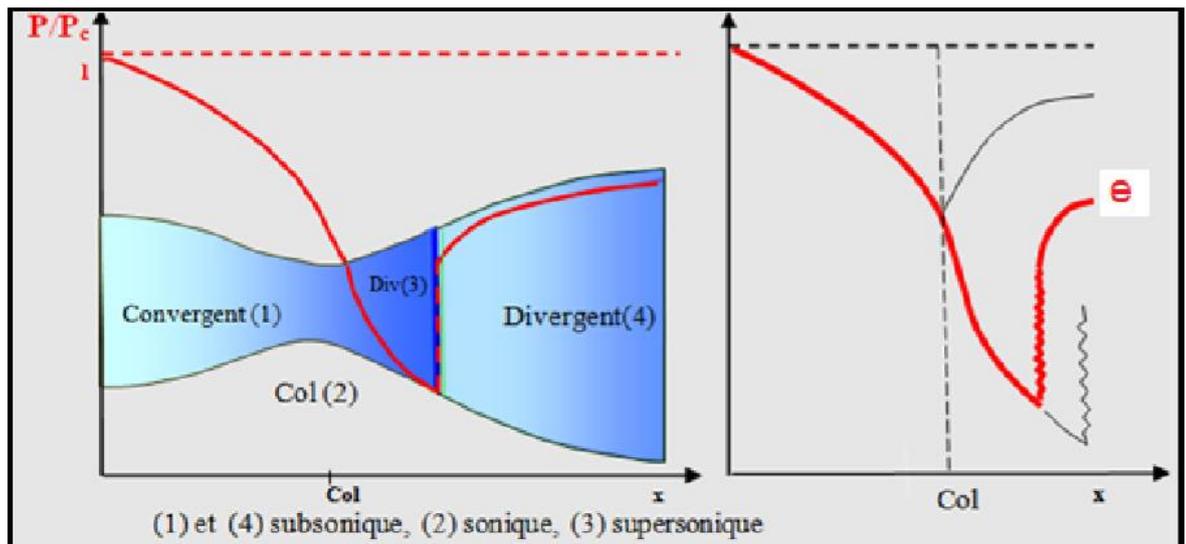


Figure 5.4 : Régime de sur-détente en 1D. [15]

L'onde de choc se déplace vers la sortie lorsque P_a diminue. On désignera par P_{ac2} la pression ambiante pour laquelle l'onde de choc se place à la sortie de la tuyère.

En diminuant encore la pression P_a au-dessous de P_{ac2} , l'écoulement dans le divergent est entièrement supersonique, la diminution de P_a ne modifie plus l'écoulement à l'intérieur de la tuyère. L'adaptation de la pression entre la sortie de la tuyère et la pression du milieu ambiant doit se faire à l'extérieur du divergent par l'intermédiaire d'ondes de choc obliques.

5.2.4. Ecoulement supersonique adapté

Si P_a continue à diminuer il arrive un moment où l'écoulement est entièrement supersonique dans toute la tuyère. Le phénomène d'onde de choc disparaît et il n'y a plus de recompression externe, alors la tuyère est parfaitement adaptée. L'évolution du gaz est représentée par la courbe (f) de la figure 5.5. Désignons par P_{ac3} la pression ambiante pour laquelle ce phénomène se produit. P_{ac3} est appelé pression d'adaptation.

5.2.5. Ecoulement supersonique non adapté

Lorsque la pression P_a devient inférieure à P_{ac3} l'écoulement est dit supersonique non adapté. L'écoulement dans la tuyère de Laval ne subit aucune modification, l'adaptation entre la tuyère et la pression ambiante doit se faire à l'extérieur de la tuyère par l'intermédiaire d'une série d'ondes de détente obliques.

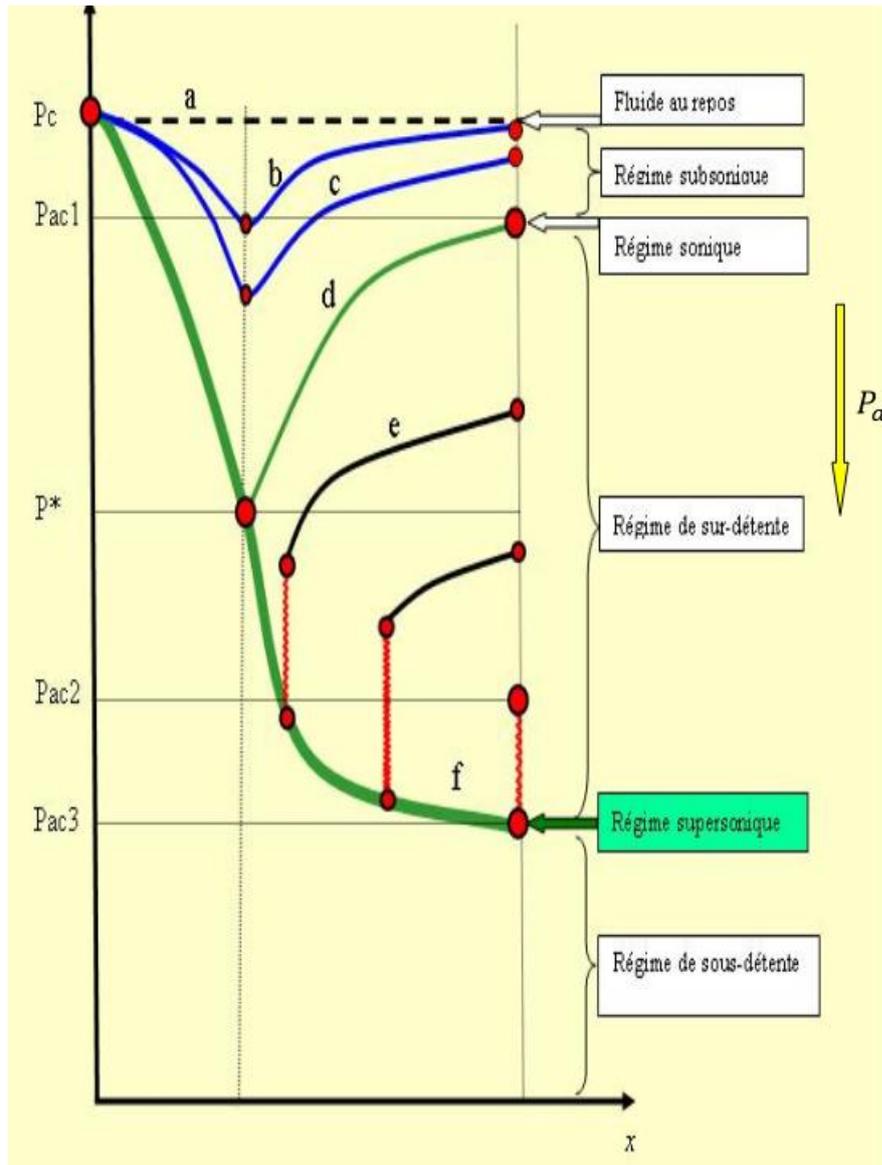


Figure 5.5 : Les régimes d'écoulement dans une tuyère supersonique. Evolution de la pression le long de la tuyère. [15]

5.3. Présentation du problème

Notre problème réside dans la comparaison de notre simulation avec la théorie. Pour cela, on a choisi de faire le cas d'un écoulement supersonique adapté sans onde de choc interne. L'écoulement est par la suite isentropique en amont et en aval du col.

Les conditions de notre tuyère :

Considérons le fluide dans le réservoir comme étant un gaz parfait entièrement au repos, le rapport de section est $\frac{A_s}{A^*} = 4.235$ ou A_s est la section de sortie du divergeant. On cherche à trouver les paramètres thermodynamiques au col et à la sortie de la tuyère.

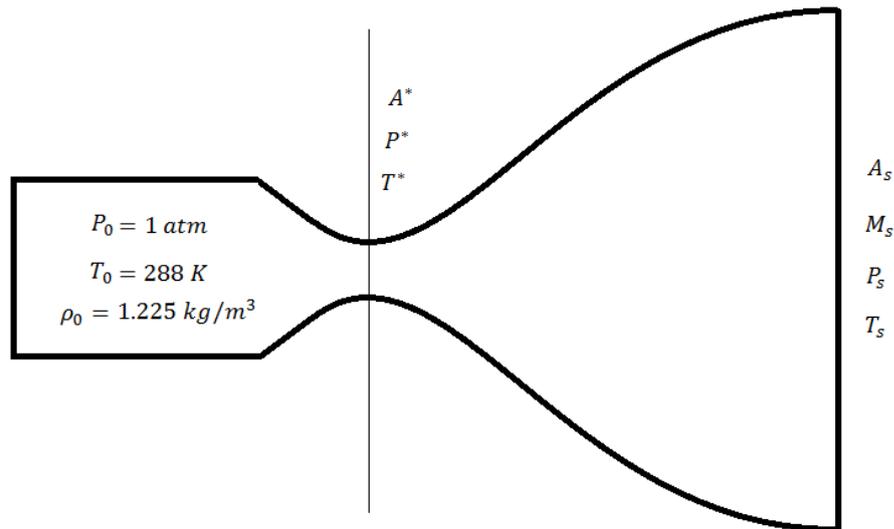


Figure 5.6 : Schémas du problème.

Solution analytique :

D'après la table des écoulements isentropique [10], $\frac{A_s}{A^*} = 4,235$ correspond à un Mach de sortie $M_s = 3$.

Au col $M^* = 1$,

$$P^* = \frac{P^*}{P_0} P_0 = 0,528 \cdot 1 = 0,528 \text{ atm} = 53499,6 \text{ Pa}$$

$$T^* = \frac{T^*}{T_0} T_0 = 0,833 \cdot 288 = 240 \text{ K}$$

$$U^* = M^* \sqrt{\gamma r T^*} = 1 \cdot \sqrt{1,4 \cdot 287 \cdot 240} = 310,64 \text{ m/s}$$

A la sortie $M_s = 3$,

$$P_s = \frac{P_s}{P_0} P_0 = \frac{1}{36,73} \cdot 1 = 0,027225 \text{ atm} = 2758,64 \text{ Pa}$$

$$T_s = \frac{T_s}{T_0} T_0 = \frac{1}{2,8} \cdot 288 = 102,857 \text{ K}$$

$$U_s = M_s \sqrt{\gamma r T_s} = 3 \cdot \sqrt{1,4 \cdot 287 \cdot 102,857} = 609,878 \text{ m/s}$$

La suite de ce chapitre traite des simulations numériques d'un groupe de tuyère qui a le même rapport de section, le même nombre de Mach de sortie, mais différents type de géométries à l'intérieur.

5.4. Prétraitement

Il s'agit donc d'une simulation numérique de plusieurs formes de divergeant de tuyère supersonique. On a choisi de faire la simulation uniquement sur le divergeant puisque c'est la partie la plus importante dans la tuyère pour gagner un peu de temps de calcul.

5.4.1. Génération des maillages

Pour la génération de maillage on a choisi l'utilitaire de maillage blockMesh. Parceque la saisie du programme blockMeshDict est très long et peut engendrer des fautes et pour gagner du temps et éviter les erreurs, on a créé un programme FORTRAN qui a pour but de convertir le fichier .DAT des géométries. Tout d'abord on commence par la lecture du fichier .DAT et l'ouverture d'un fichier appeler blockMeshDict. En respectant la mise en forme du fichier blockMeshDict, on constitue un entête et les points constituant les volumes élémentaires. Ensuite, on définit les blocs qui constituent notre géométrie, chaque 8 points constitue un bloc. Enfin, on détermine les liaisons entre ces points par des interpolations linéaires.

```

real x(7700),y(7700),z(7700)
integer r(800),j,k
open (1,file=/home/med/OpenFOAM/med-2.3.0/run/Projet/carrÃ©e/D_3D_GP.DAT')
open (2,file=/home/med/OpenFOAM/med-
2.3.0/run/Projet/carrÃ©e/GP2/constant/polyMesh/blockMeshDict')
do i=0,7699
read(1,*)x(i),y(i),z(i)
enddo
do k=1,100
do i=k*77-(k),7699
x(i)=x(i+1)
y(i)=y(i+1)
z(i)=z(i+1)
enddo
enddo
write(2,3)/*-----*- C++ -*-----*/
write(2,3)|=====|
write(2,3)| \ / F ield      | OpenFOAM: The Open Source CFD Toolbox      |"
write(2,3)| \ / O peration  | Version: 2.3.0                          |"
write(2,3)| \ / A nd        | Web:   www.OpenFOAM.org                  |"
write(2,3)| \ \ M anipulation |                                         |"
write(2,3)/*-----*/
write(2,3)"FoamFile"
write(2,3){
write(2,3)"  version  2.0;"
write(2,3)"  format   ascii;"
write(2,3)"  class    dictionary;"
write(2,3)"  object   blockMeshDict;"
write(2,3)}
write(2,3)// ***** //"
write(2,3)"convertToMeters 1;"
write(2,3)"vertices"
write(2,3)"(
3 format(80(a))
k=0
do i=0,7599
if (y(i).eq.-z(i))then
r(k)=i
k=k+1
endif
if (y(i)*y(i-1).lt.0)then
r(k)=i
k=k+1
endif
if (y(i).eq.z(i))then
r(k)=i
k=k+1
endif
if (z(i)*z(i-1).lt.0)then
r(k)=i
k=k+1
endif

```

```

endif
enddo
do i=0,799
if(abs(y(r(i))).lt.0.08)then
y(r(i))=0
endif
if(abs(z(r(i))).lt.0.08)then
z(r(i))=0
endif
write(2,*)("x(r(i)),y(r(i)),z(r(i)),")","/",i
enddo
do i=0,7599,76
write(2,*)("x(i),0,0)","/",j+800
j=j+1
enddo
write(2,*)";"
write(2,*)"blocks"
write(2,*)("
k=0
do i=0,799-15,8
if(i.gt.799-15-12)then
write(2,*)"hex(",k+800,i+5,i+6,i+7, k+801,i+13,i+14,i+15,") (20 20 7) simpleGrading (.1 .1 1)"
write(2,*)"hex(",i+1,k+800,i+7,i, i+9,k+801,i+15,i+8,") (20 20 7) simpleGrading (10 .1 1)"
write(2,*)"hex(",i+2,i+3,k+800,i+1, i+10,i+11,k+801,i+9,") (20 20 7) simpleGrading (10 10 1)"
write(2,*)"hex(",i+3,i+4,i+5,k+800, i+11,i+12,i+13,k+801,") (20 20 7) simpleGrading (.1 10 1)"
else
write(2,*)"hex(",k+800,i+5,i+6,i+7, k+801,i+13,i+14,i+15,") (20 20 1) simpleGrading (.1 .1 1)"
write(2,*)"hex(",i+1,k+800,i+7,i, i+9,k+801,i+15,i+8,") (20 20 1) simpleGrading (10 .1 1)"
write(2,*)"hex(",i+2,i+3,k+800,i+1, i+10,i+11,k+801,i+9,") (20 20 1) simpleGrading (10 10 1)"
write(2,*)"hex(",i+3,i+4,i+5,k+800, i+11,i+12,i+13,k+801,") (20 20 1) simpleGrading (.1 10 1)"
endif
k=k+1
enddo
write(2,*)";"
k=0
write(2,*)"edges"
write(2,*)("
do i=0,7590
if(i.eq.r(k))then
if((k+1)/8..eq.int((k+1)/8.))then
write(2,*)"spline",k,"",k-7,("
else
write(2,*)"spline",k,"",k+1,("
endif
do j=r(k)+1,r(k+1)-1
write(2,*)("x(j),y(j),z(j),")"
enddo
write(2,*)")"
k=k+1
endif
enddo
write(2,*)"spline",k,"",k-7,("
do j=r(k)+1,7598
write(2,*)("x(j),y(j),z(j),")"
enddo
write(2,*)")"
write(2,*)";"
write(2,*)"boundary"
write(2,*)("
write(2,*)";"
write(2,*)"mergePatchPairs"
write(2,*)("
write(2,*)";"
write(2,3)//"*****"
stop
end

```

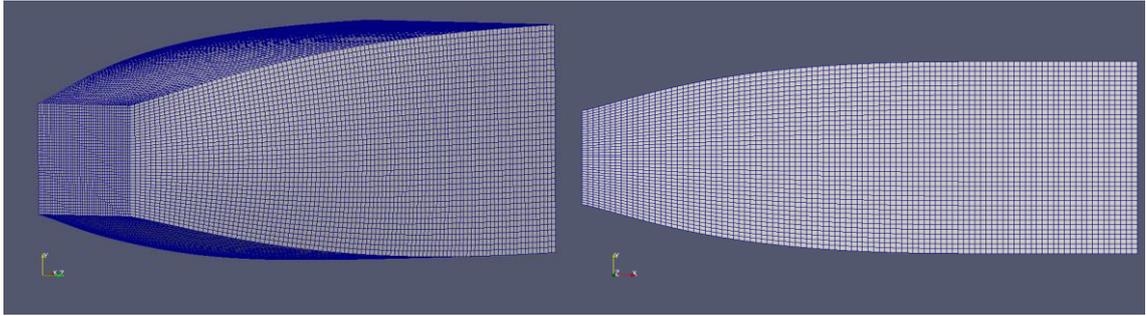


Figure 5.7 : Maillage d'une tuyère carré par blockMesh.

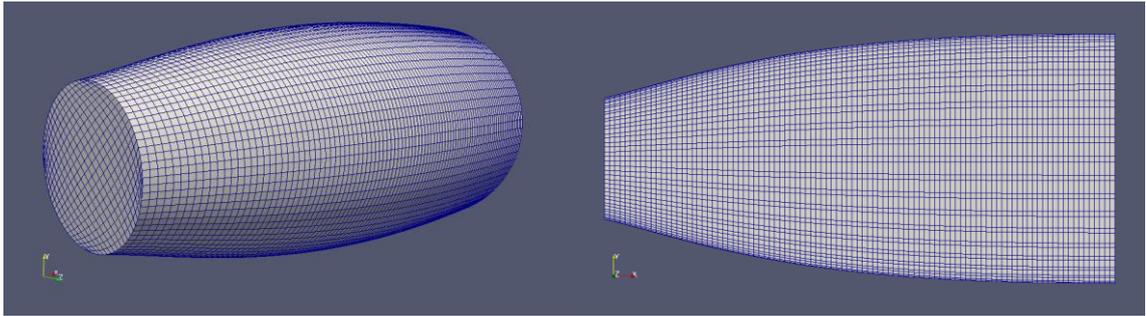


Figure 5.8 : Maillage d'une tuyère circulaire par blockMesh.

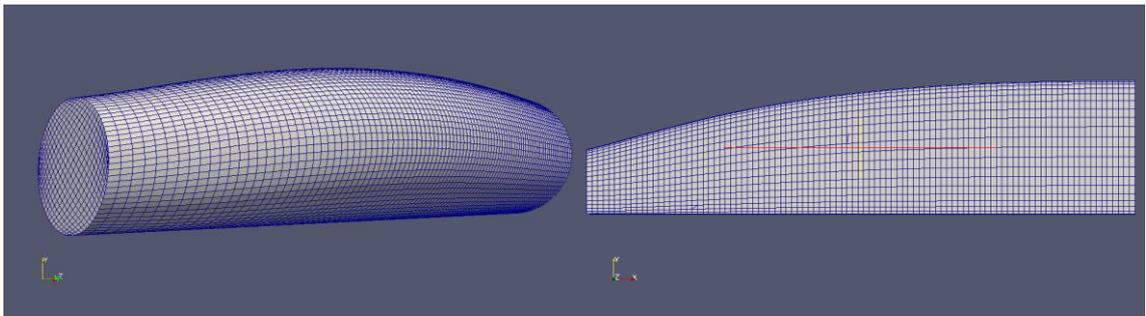


Figure 5.9 : Maillage d'une tuyère circulaire non axis-symétrique par blockMesh.

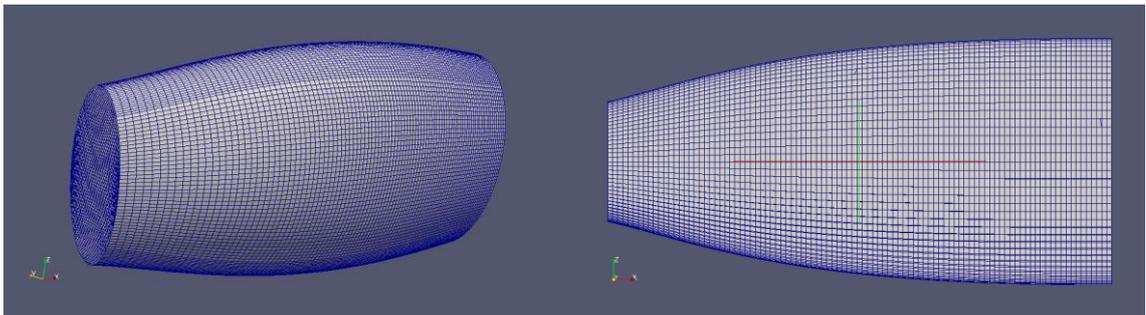


Figure 5.10 : Maillage d'une tuyère elliptique par blockMesh.

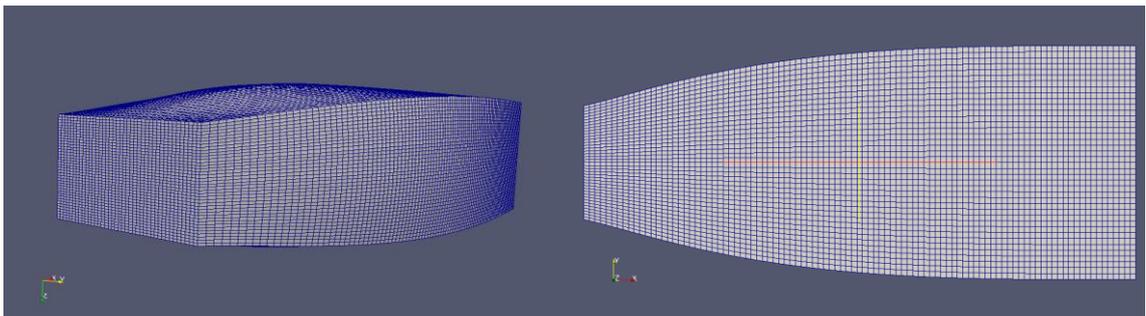


Figure 5.11 : Maillage d'une tuyère rectangulaire par blockMesh.

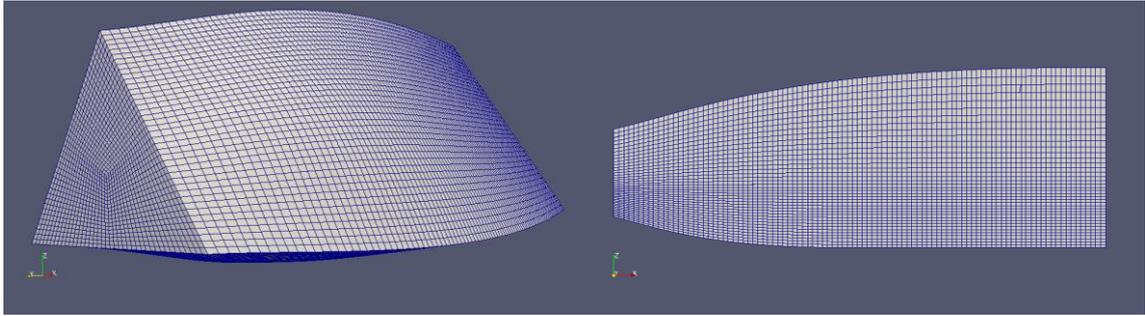
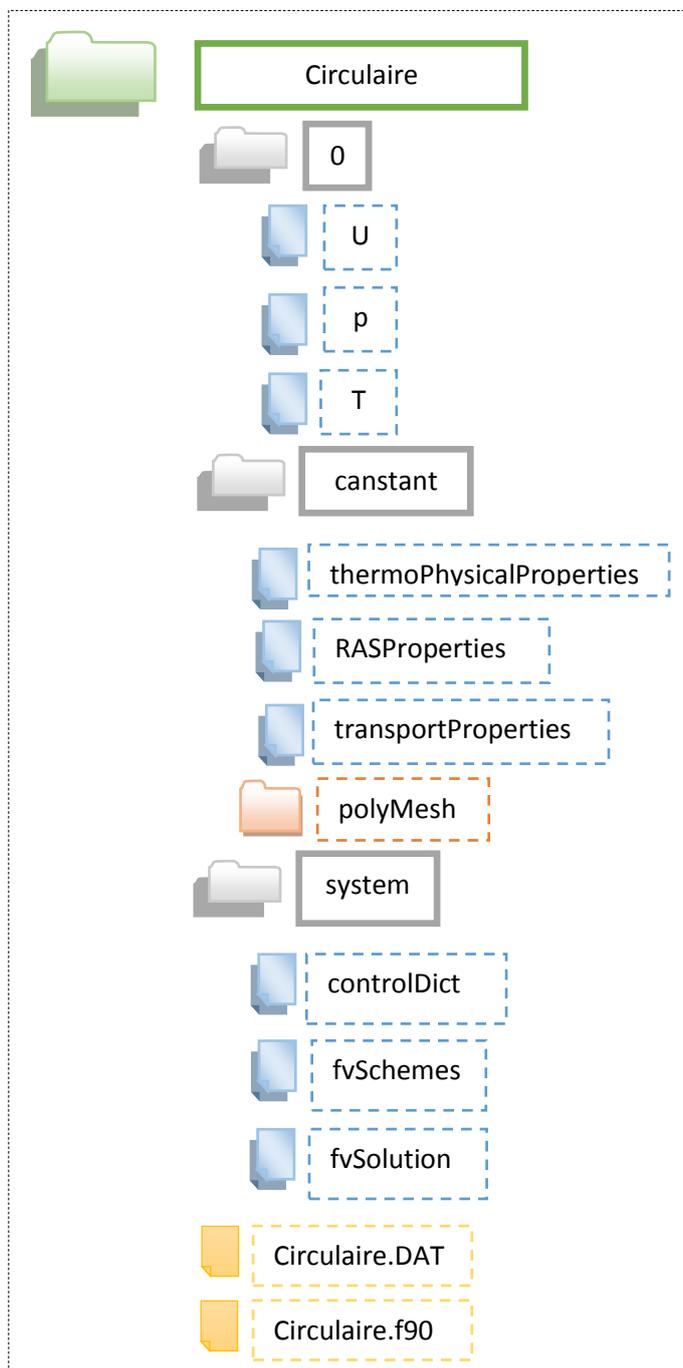


Figure 5.12 : Maillage d'une tuyère triangulaire par blockMesh.

Structure du cas :



5.5. Traitement

Le solveur utilisé dans cette simulation est rhoCentralFoam [16] qui est un solveur basé sur la masse volumique, c'est un solveur destiné aux calculs des écoulements compressibles suivant les schémas de Kurganov et Tadmor [17]. C'est une méthode de discrétisation qui peut être utilisé sur les scalaires et sur les vecteurs. L'algorithme est basé sur les différences centrales d'où vient le mot rhoCentralFoam. Ce solveur ressoude les équations de Navier Stokes et l'équation d'énergie.

Conditions aux limites :

Tableau 5.1 : Conditions aux limites.

Paramètres	Faces	Conditions aux limites	Description
U	throat	fixedValue, uniform (310,64, 0 0)	Vitesse uniforme 310,64m/s fixé.
	Outlet	inletOutlet, value 609,88	Ecoulement libre avec estimation de la vitesse de sortie.
	walls	zeroGradient	Conditions de non glissement.
p	throat	fixedValue, uniform 53499,6	Pression fixé à l'entrée
	Outlet	fixedValue, uniform2758,64	Pression estimé à la sortie
	Walls	zeroGradient	Gradient nulle dans la direction de la normale.
T	Throat	fixedValue, uniform240	Température fixe a l'entrée
	Outlet	fixedValue, uniform102,86	Estimation de la température a la sortie
	Walls	zeroGradient	Paroi adiabatique

5.6. Post-traitement

Les résultats obtenu son visualiser par Paraview. La tuyère circulaire axisymétrique, comme étant la plus utilisé, est prise comme référence.

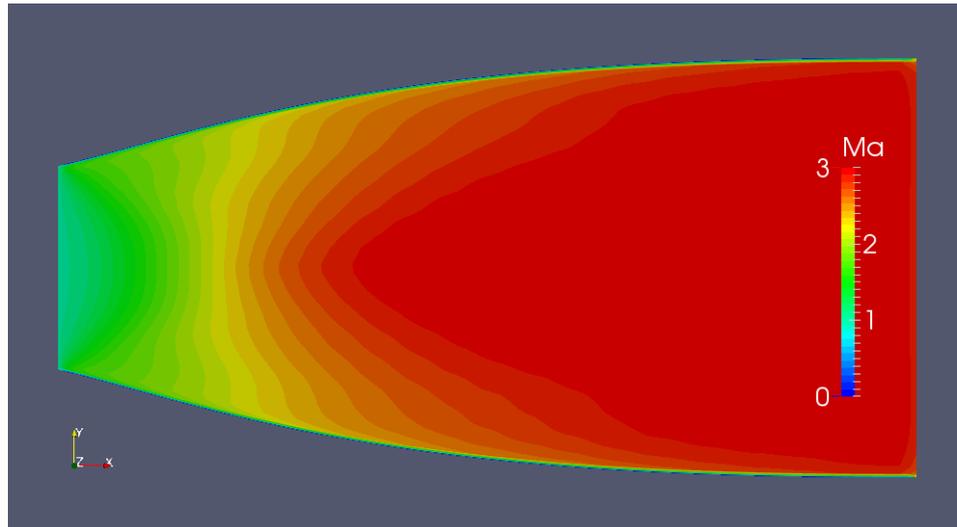


Figure 5.13 : Champs des iso-Mach dans la tuyère axisymétrique.

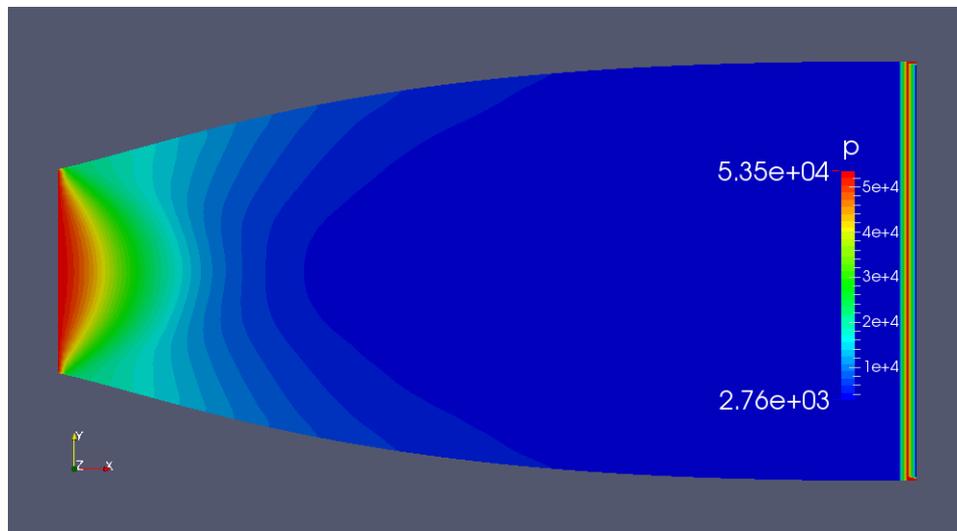


Figure 5.14 : Champs des iso-bar dans la tuyère axisymétrique.

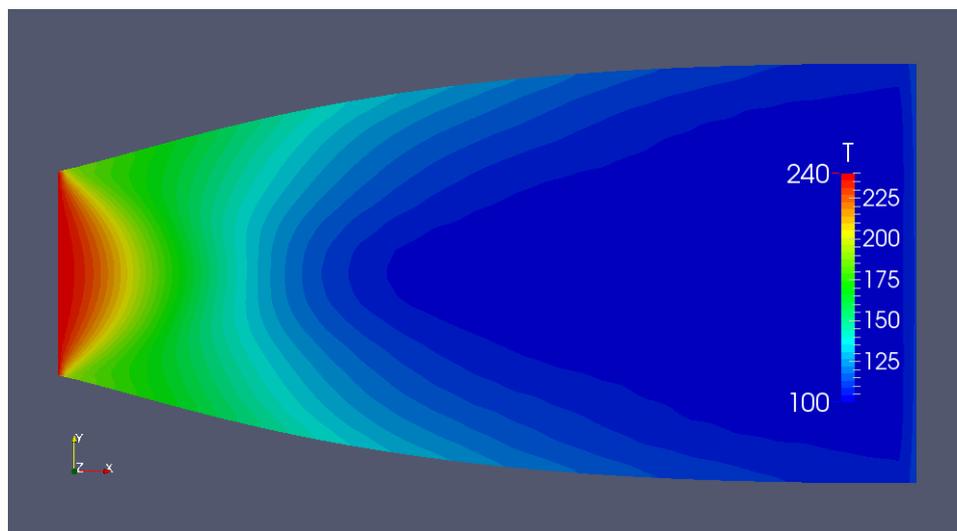


Figure 5.15 : Champs des isothermes dans la tuyère axisymétrique.

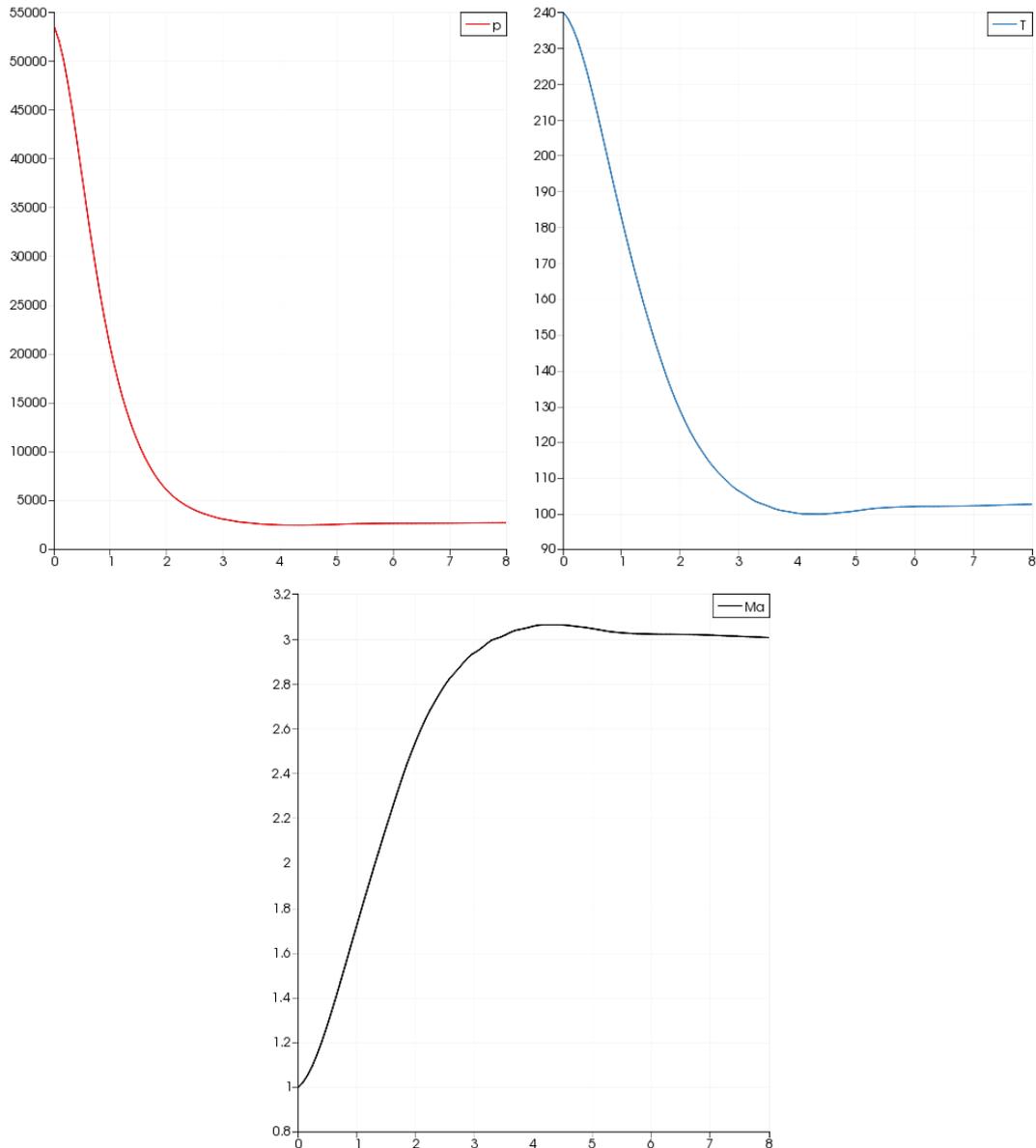


Figure 5.16 : Courbe des évolutions des paramètres thermodynamiques pour la tuyère axisymétrique.

Les résultats de simulation obtenus concordent très bien avec les calculs analytiques et les évolutions sont logiques. Les contours des pressions et de nombre de Mach à l'intérieur de la tuyère sont comme prévu.

5.6.1. Différents résultats

Chaque type de tuyère a donné presque le même nombre de Mach à la sortie, mais l'allure de ces paramètres thermodynamiques change considérablement à l'intérieur.

5.6.1.1. Comparaison des paramètres thermodynamiques entre différentes géométries

Les résultats obtenus permettent la comparaison entre différents types de géométrie de tuyère en comparant les champs des grandeurs thermodynamiques.

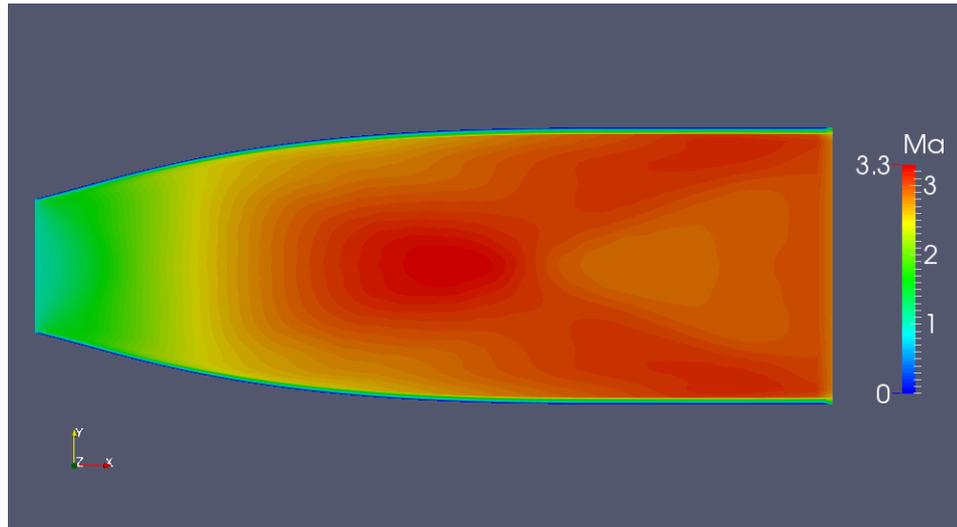


Figure 5.17 : Champs des iso-Mach dans la tuyère carrée.

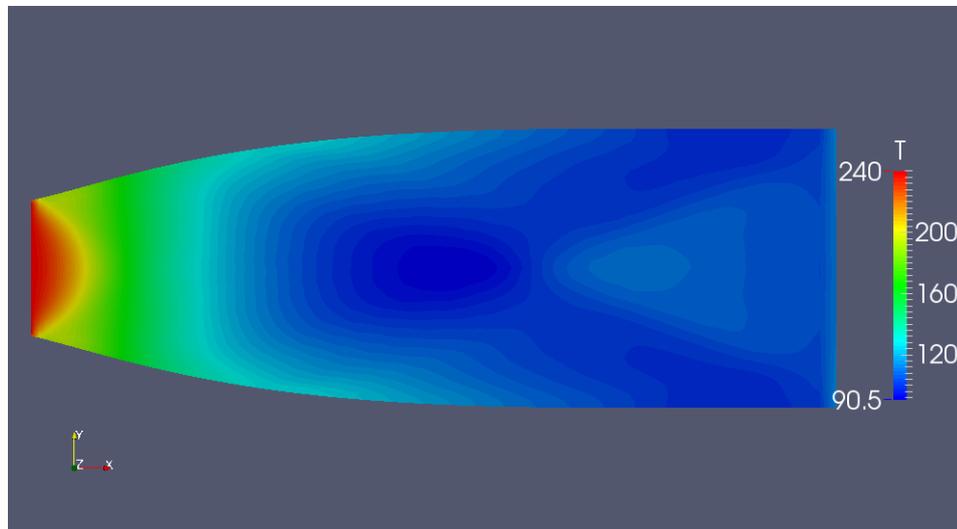


Figure 5.18 : Champs des isothermes dans la tuyère carrée.

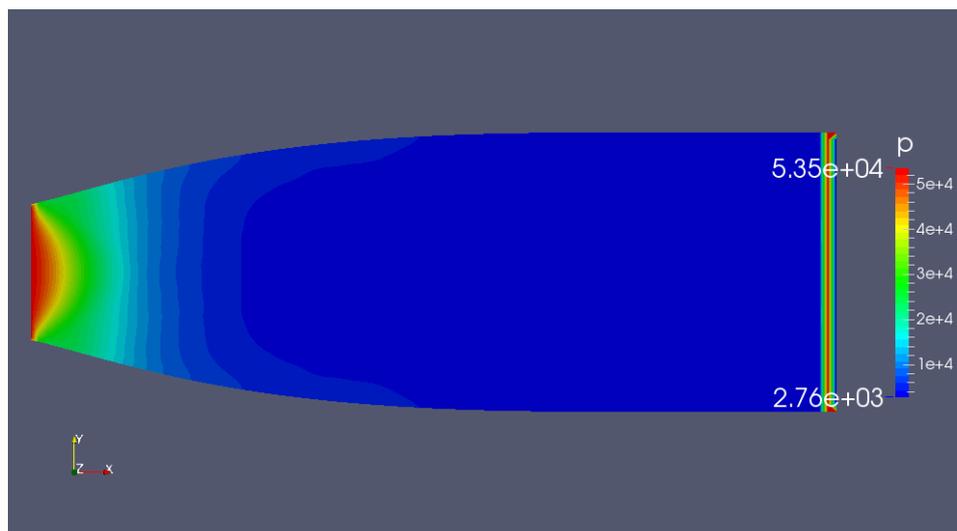


Figure 5.19 : Champs des isobares dans la tuyère carrée.

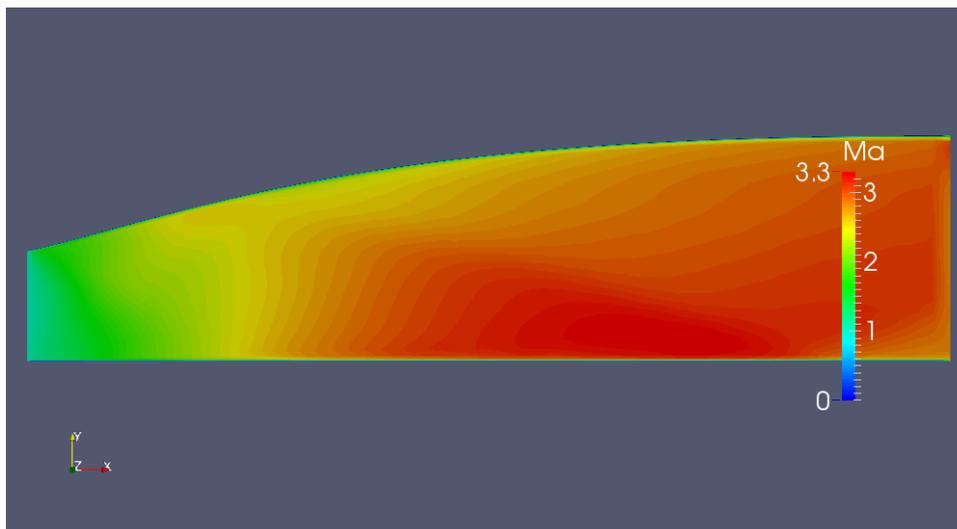


Figure 5.20 : Champs des iso-Mach dans la tuyère circulaire non axisymétrique.

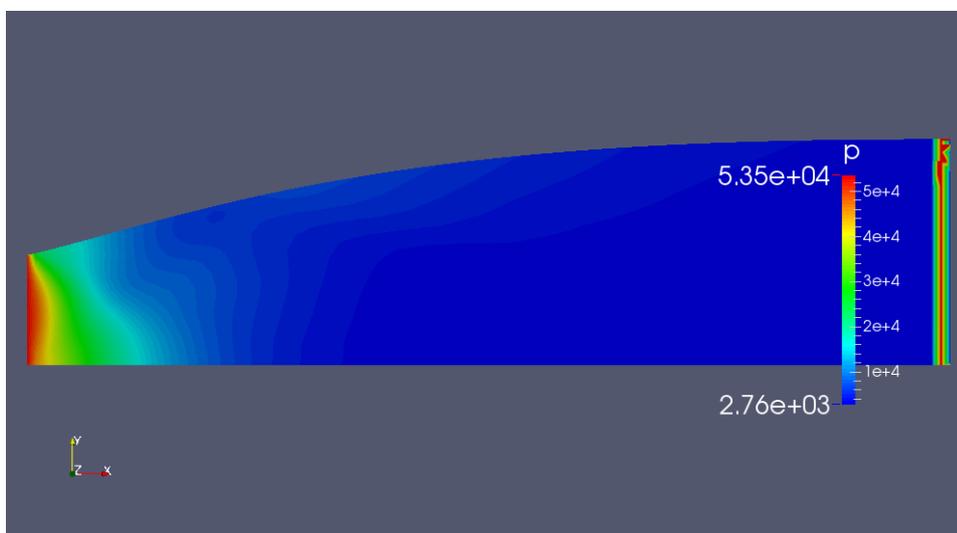


Figure 5.21 : Champs des isobares dans la tuyère circulaire non axisymétrique.

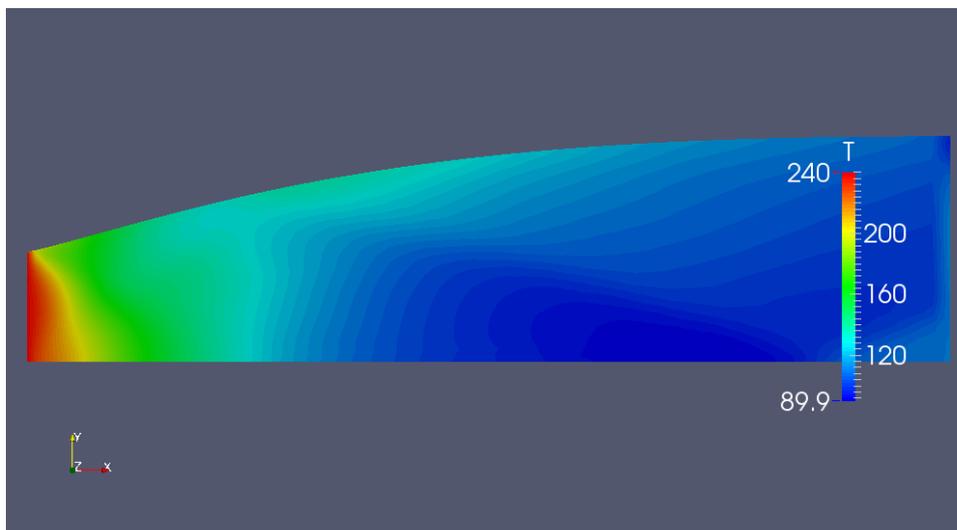


Figure 5.22 : Champs des isothermes dans la tuyère circulaire non axisymétrique.

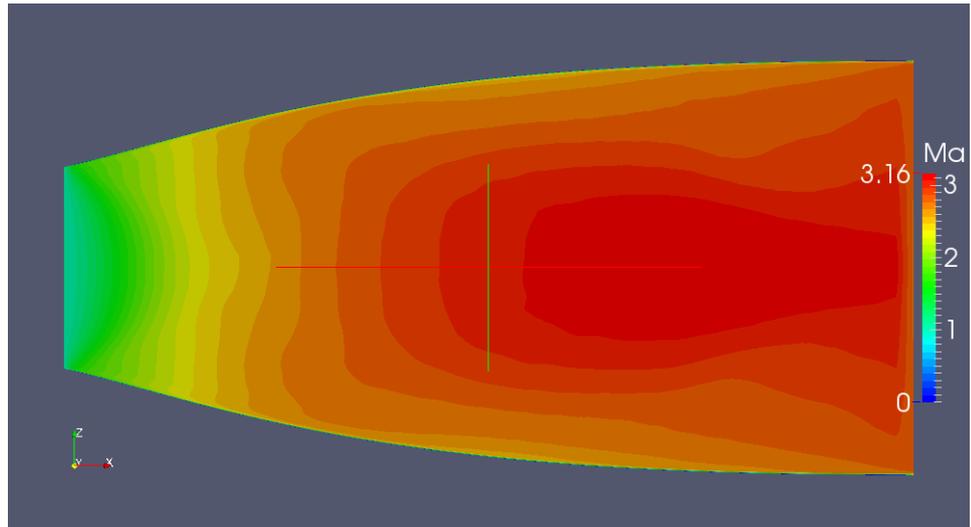


Figure 5.23 : Champs des iso-Mach dans la tuyère circulaire elliptique.

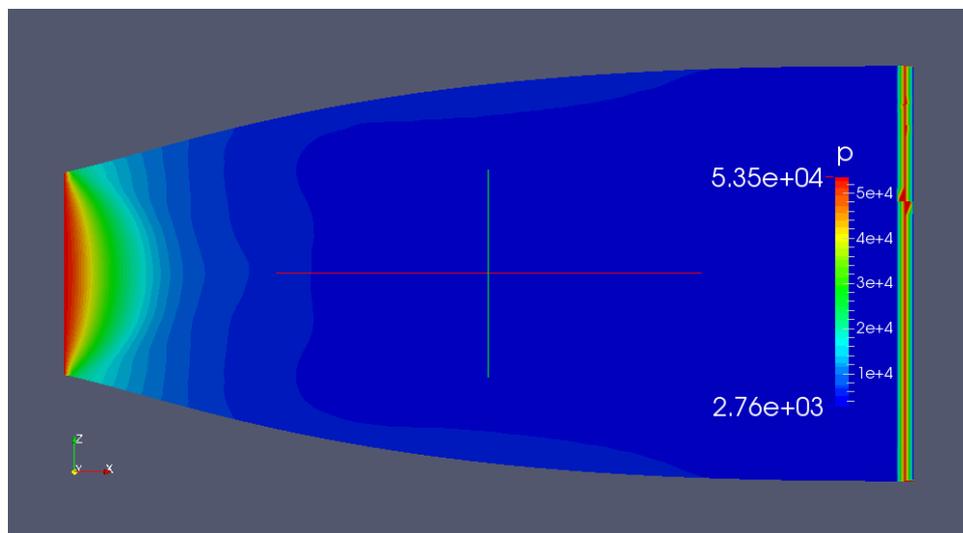


Figure 5.24 : Champs des isobares dans la tuyère circulaire elliptique.

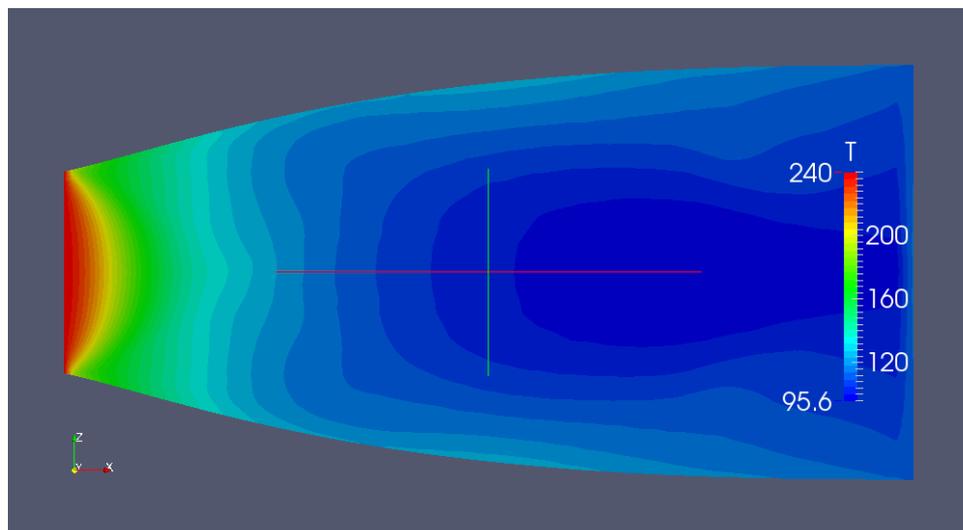


Figure 5.25 : Champs des isothermes dans la tuyère elliptique.

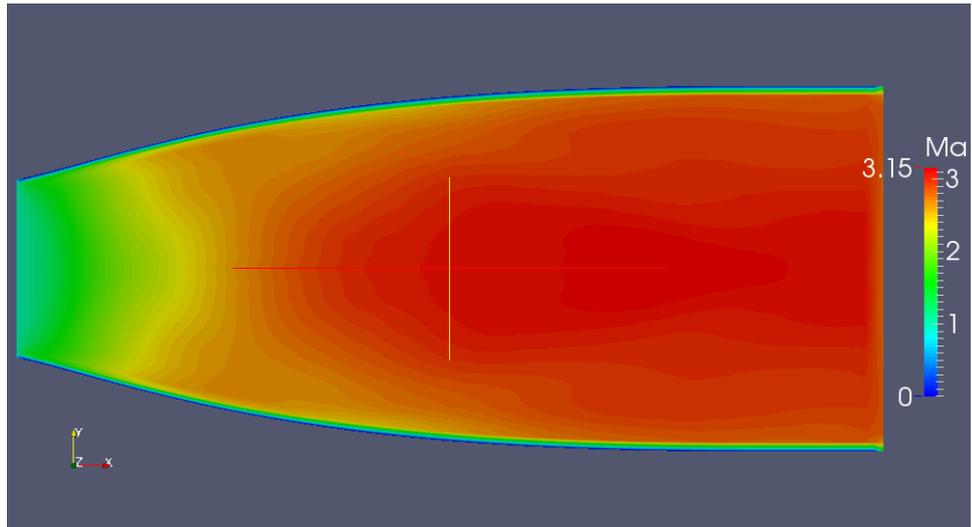


Figure 5.26 : Champs des iso-Mach dans la tuyère rectangulaire.

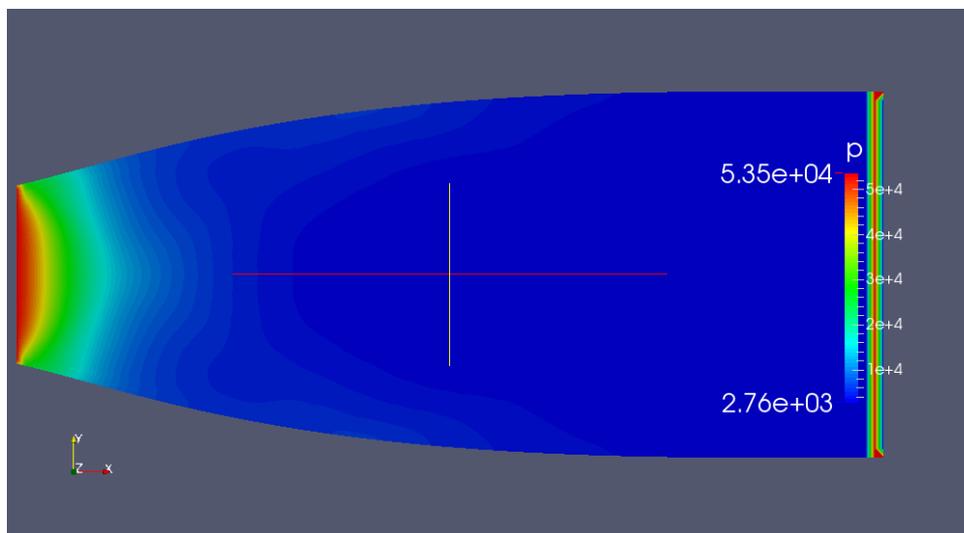


Figure 5.27 : Champs des isobares dans la tuyère rectangulaire.

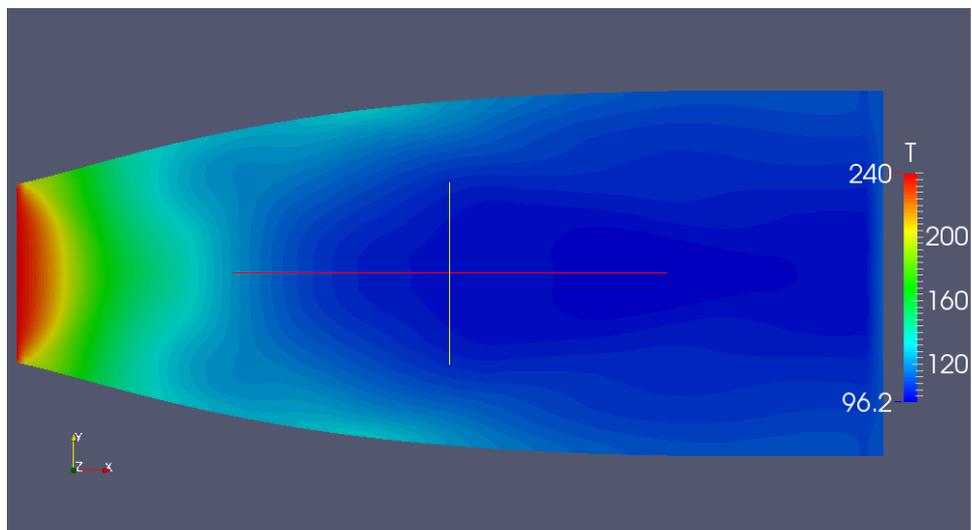


Figure 5.28 : Champs des isothermes dans la tuyère rectangulaire.

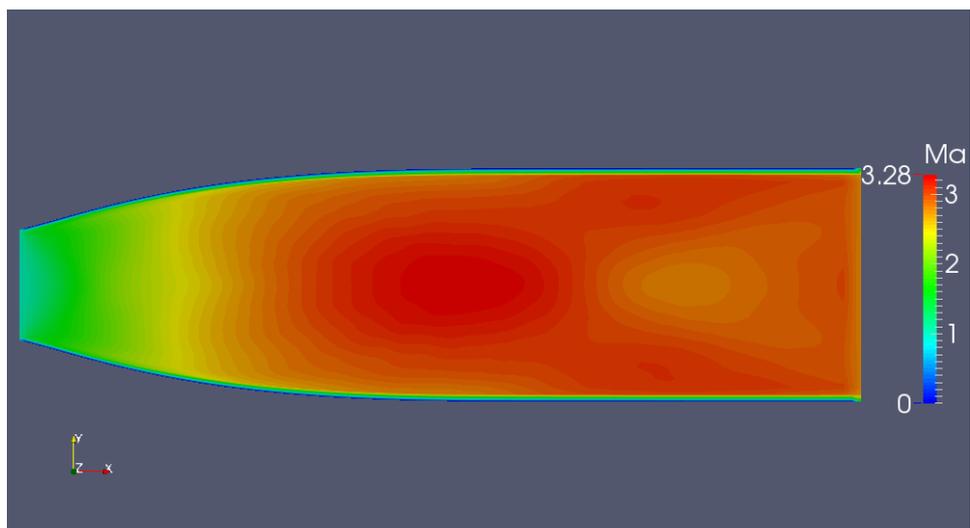


Figure 5.29 : Champs des iso-Mach dans la tuyère triangulaire.

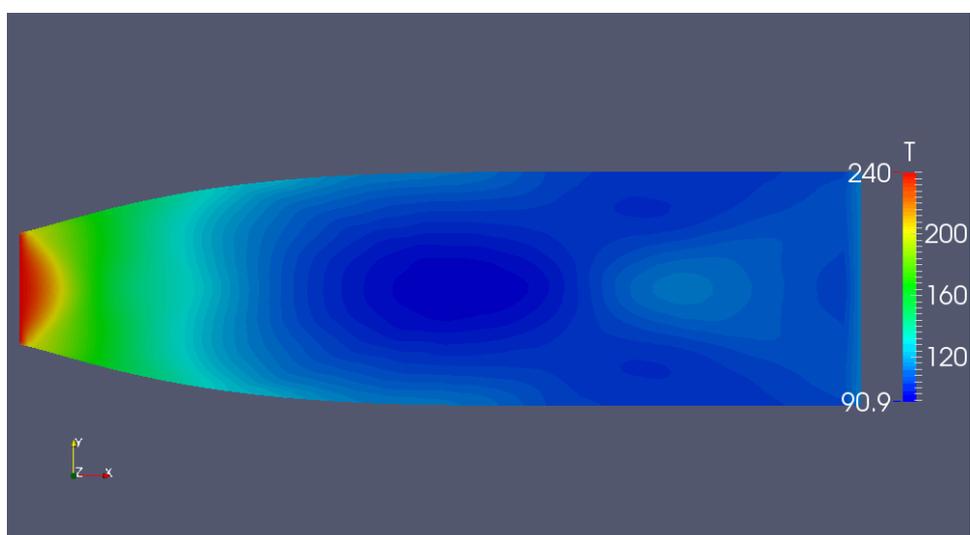


Figure 5.30 : Champs des isothermes dans la tuyère triangulaire.

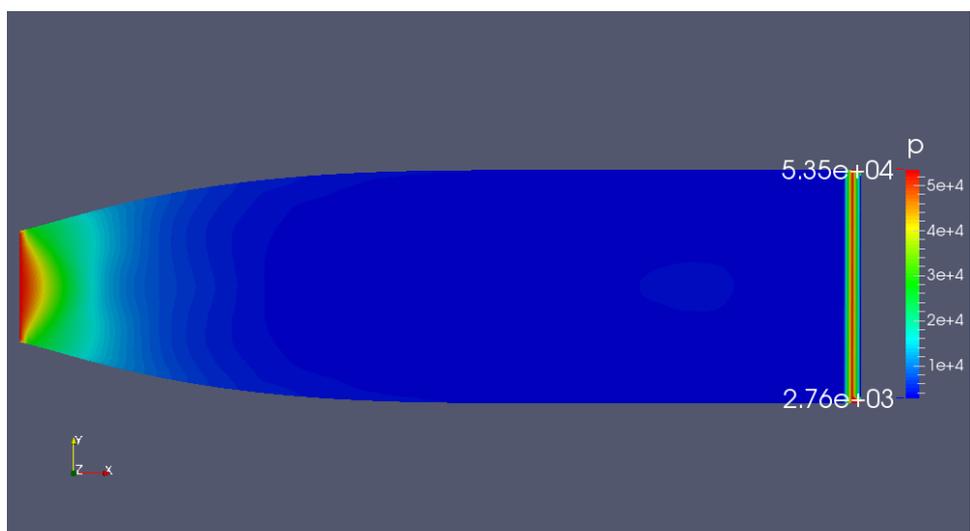


Figure 5.31 : Champs des isobares dans la tuyère triangulaire.

Comparaison des iso-Mach à l'intérieur des tuyères en vue de section :

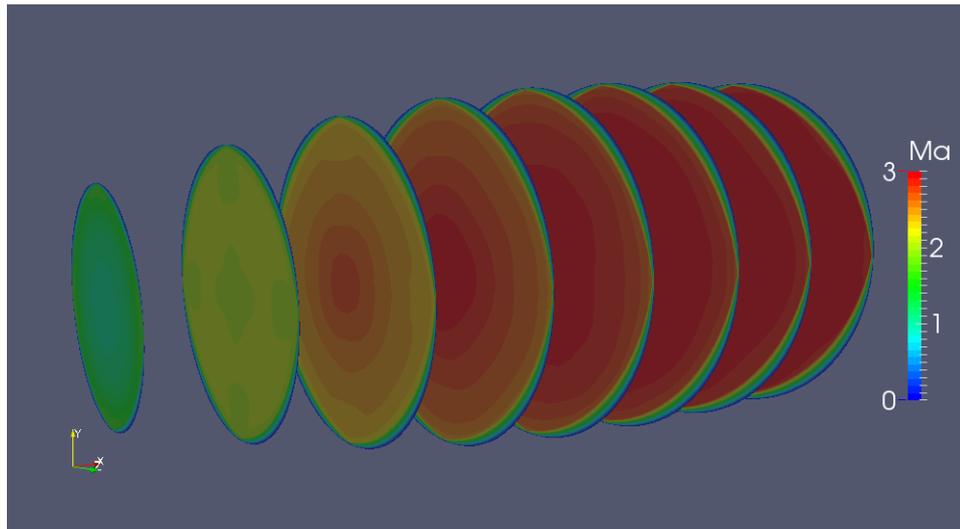


Figure 5.32 : Champs des iso-Mach dans la tuyère circulaire.

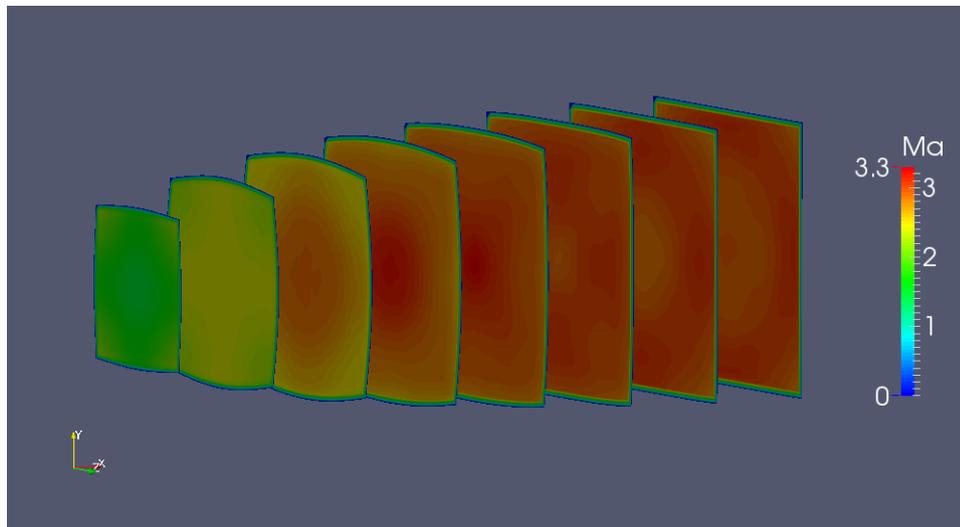


Figure 5.33 : Champs des iso-Mach dans la tuyère carrée.

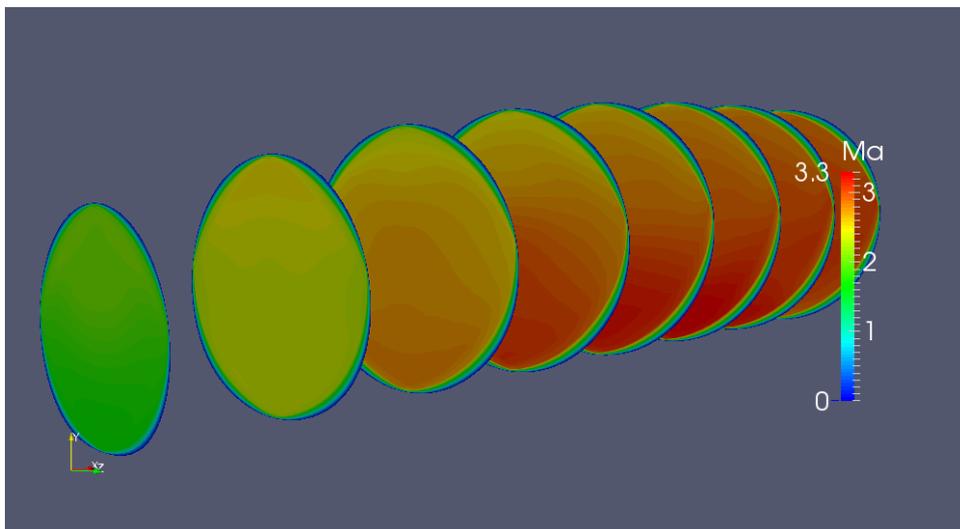


Figure 5.34 : Champs des iso-Mach dans la tuyère circulaire non axisymétrique.

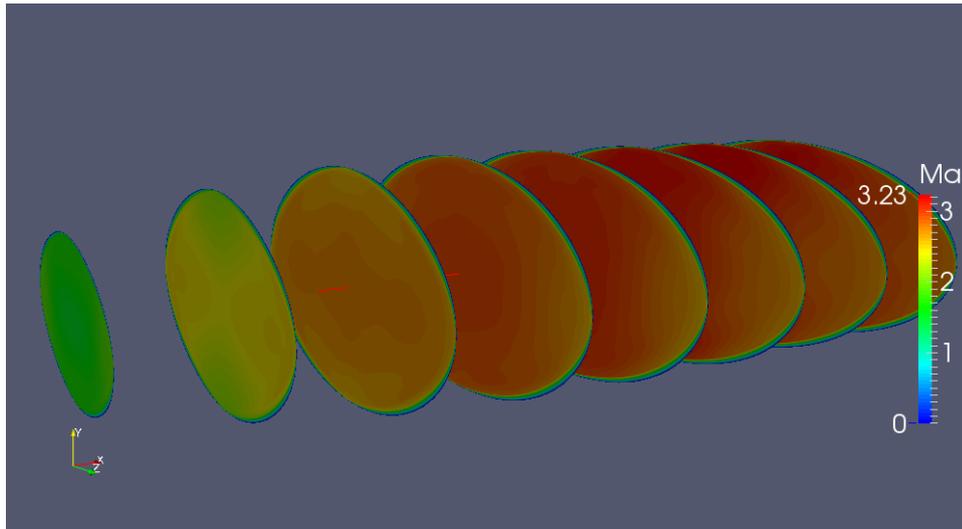


Figure 5.35 : Profil des iso-Mach dans la tuyère elliptique.

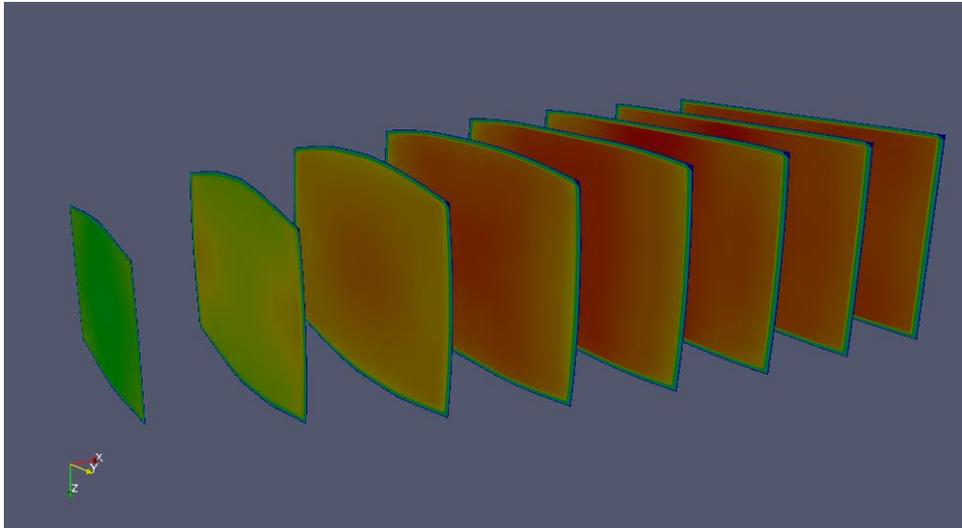


Figure 5.36 : Profil des iso-Mach dans la tuyère carrée.

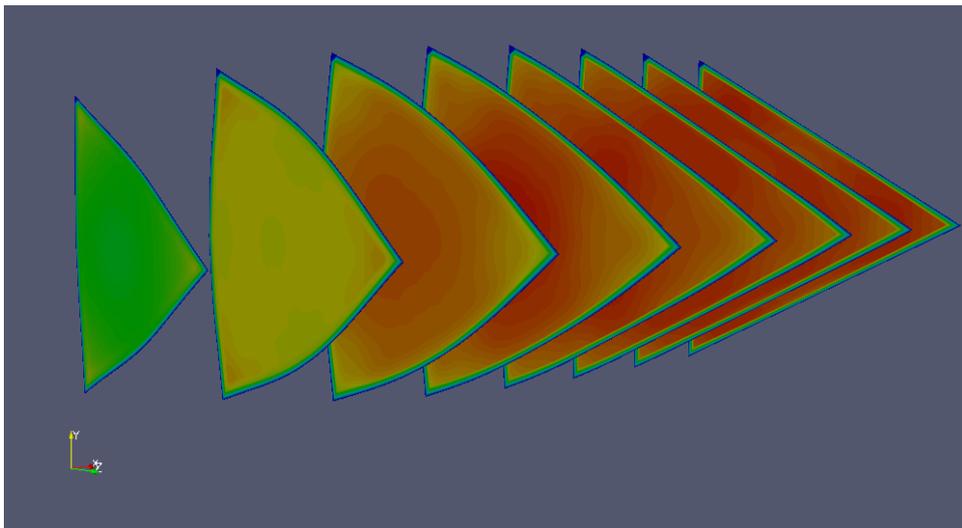


Figure 5.37 : Profil des iso-Mach dans la tuyère triangulaire.

5.7. Conclusion

En théorie, la valeur de la section d'une tuyère influe sur l'écoulement comme l'indiquent les lois de conservation mais ils ne portent aucune condition sur la forme de section. Mais en réalité la géométrie de cette section influe sur les champs des caractéristiques thermodynamiques dans la tuyère comme la valeur exacte du volume occupé par le fluide change.

Les maillages générés par le programme blockMesh sont très précis et bien structurés mais ils peuvent être très longs à mettre en place ce qui engendre une grande perte de temps et d'éventuelles erreurs. Seulement, l'établissement de ce programme peut être automatisé par un autre programme externe écrit en FORTRAN afin de gagner de temps et d'éviter les erreurs.

Basé sur la résolution des équations gouvernant l'écoulement en fonction de la masse volumique, le solveur rhoCentralFoam est un calculateur puissant et économise le temps. Comme tous les solveurs d'OpenFOAM, ce solveur accepte les modifications et les optimisations des utilisateurs qui ont de l'expérience dans la modélisation mathématique et les schémas numériques.

CONCLUSION GENERALE

Durant ce PFE, OpenFOAM a permis de démontrer son efficacité, en terme de résultats et de rapidité de calcul, ainsi que sa fiabilité si on le compare avec la littérature. De plus, il a été démontré qu'il était possible par son biais de mettre en place des cas d'étude de façon beaucoup plus rapide qu'avec n'importe quel autre logiciel.

Cependant, il a aussi été démontré que son utilisation pouvait être complexe et qu'elle demandait un certain savoir et une certaine expérience. Néanmoins, il a été présenté dans le cadre de ce document un résumé de ce savoir minimum que doit posséder une personne voudrait mettre en œuvre cet outil. OpenFOAM peut être aussi un outil convivial à condition que l'on s'en donne les moyens.

En conclusion, il est possible de dire qu'une intégration d'OpenFOAM en entreprise est tout à fait possible et qu'elle ouvrirait des perspectives intéressantes : possibilité d'augmenter sans limite le nombre de postes possédant un logiciel de CFD puissant et effectif ; possibilité de créer ses propres solveurs afin d'être entièrement indépendant et de posséder un outil entièrement adapté aux applications propres à l'entreprise, etc.

Ce PFE a donc permis de se rendre compte que la complexité d'OpenFOAM ne devait pas être un obstacle à son utilisation en entreprise, bien au contraire. Et, qu'elle en tirera autant de gain financier que de satisfaction en rejoignant la communauté des utilisateurs de ce merveilleux outil qu'est OpenFOAM.

PERSPECTIVES

Dans cette étude, l'utilisation d'OpenFOAM pour les simulations des écoulements internes et externes nous a donné beaucoup de perspectives et d'idées pour le futur. Ainsi, cet éminent logiciel augmente les attentes de beaucoup de gens dans différents domaines.

Ce projet se limite à l'utilisation d'OpenFOAM du point de vue du non-programmeur et ne rentre pas dans les détails de la création des solveurs, mais uniquement de l'utilisation des solveurs existants. Cependant, l'utilisateur peut aller plus loin en modifiant ou en créant ses propres solveurs et les adaptés à son cas.

Cette étude peut intéresser également d'autres secteurs que l'aérodynamique, puisqu'OpenFOAM permet en outre de faire des études de combustion, de mécanique, d'électromagnétique ou de finance et que, pour chacune de ces études, les bases de l'utilisation d'OpenFOAM sont les mêmes.

Références

- [1]. <http://www.openfoam.org/download>.
- [2]. <https://fr.wikipedia.org/wiki/OpenFOAM>.
- [3]. https://fr.wikipedia.org/wiki/Open_source.
- [4]. https://fr.wikipedia.org/wiki/Programmation_orientée_objet.
- [5]. https://fr.wikipedia.org/wiki/Licence_publique_générale_GNU.
- [6]. https://fr.wikipedia.org/wiki/Shell_Unix.
- [7]. OpenFOAM The Open Source CFD Toolbox Programmer's Guide Version 2.3.1 3rd December 2014.
- [8]. OpenFOAM The Open Source CFD Toolbox User Guide Version 2.3.1 3rd December 2014.
- [9]. Kenneth Moreland, The ParaView Tutorial Version 3.6, Sandia National Laboratories.
- [10]. John D. Anderson, FUNDAMENTALS OF AERODYNAMICS Third Edition, McGraw-Hill 2001.
- [11]. Robert M. Pinkerton, NACA Report No 563, NASA.
- [12]. <https://searchcode.com/codesearch/view/96758741/>.
- [13]. <http://www.openfoam.org/features/mesh-motion.php>.
- [14]. <https://fr.wikipedia.org/wiki/Tuyère>.
- [15]. Thèse de doctorat, Mahfoudi El-Ahcene, CONTRIBUTION A L'ETUDE DES PROFILS DE TUYERES EN ECOULEMENTS SUPERSONIQUES VISQUEUX PAR LA METHODE DES VOLUMES FINIS, Mars 2014.
- [16]. <http://www.openfoam.org/features/standard-solvers.php>.
- [17].
https://en.wikipedia.org/wiki/MUSCL_scheme#Kurganov_and_Tadmor_central_scheme.

LISTE DES FIGURES

Figure 1.1 : Structure basique d'un dossier de cas sur OpenFOAM.....	9
Figure 1.2 : Exemple d'dossier temporelle « 0 »	10
Figure 1.3 : Exemple d'un dossier « constant ».....	11
Figure 1.4 : Exemple d'un dossier « system ».....	12
Figure 1.5 : Entête du fichier blockMeshDict.....	13
Figure 1.6 : Ligne d'ouverture d'un fichier sur OpenFOAM	13
Figure 1.7 : Exemple d'un corps de fichier blockMeshDict associé à un programme en C.....	14
Figure 1.8 : Ligne de fermeture d'un fichier sur OpenFOAM.....	14
Figure 1.9 : Maillage fait par blockMesh.....	15
Figure 1.10 : Logique de la génération des blocs sur blockMesh.....	16
Figure 1.11 : Tuyère triangulaire composé de 3x100 blocs quadrilatéral avec raffinement des parois.....	17
Figure 1.12 : Un maillage autour d'une moto avec snappyHexMesh.....	19
Figure 1.13 : Présentation des filtres sur Paraview.....	23
Figure 1.14 : Importation du maillage et les paramètres désiré.....	24
Figure 1.15 : Le dossier « tutorials » sous OpenFOAM-2.3.0.....	26
Figure 1.16 : Le système des fichiers du dossier « flange ».....	27
Figure 1.17 :L'affichage des résiduels sur le terminal.....	29
Figure 1.18 : L'onglet « Properties » sur Paraview.....	30
Figure 1.19 : Visualisation du maillage.....	30
Figure 1.20 : Résultat final de la simulation.....	31
Figure 2.1 : La variation de C_x pour un cylindre circulaire en fonction de Re	34
Figure 2.2 : Régimes d'écoulements visqueux autour d'un cylindre circulaire....	35
Figure 2.3 : Écoulement de l'eau autour d'un cylindre circulaire pour $Re = 1.54$..	36
Figure 2.5 : Allée de Von Karman derrière un cylindre où $Re = 26$	36
Figure 2.5 : Allée de Von Karman derrière un cylindre où $Re = 140$	37
Figure 2.6 : Schéma du problème.....	39
Figure 2.7 : Structure du cas étudié.....	40
Figure 2.8 : Représentation des zones de maillages.....	40

Figure 2.9 : Visualisation du maillage.....	44
Figure 2.10 : Visualisation du champ de vitesse et l'allée de Von Karaman.....	46
Figure 2.11 : Évolution du coefficient de portance.....	46
Figure 2.12 : Évolution du coefficient de traîné.....	47
Figure 2.13 : Le nombre de Strouhal en fonction du nombre de Reynolds	47
Figure 2.14 : Visualisation des lignes de courants à $Re = 2$	48
Figure 2.15 : Visualisation des lignes de courants à $Re = 20$	49
Figure 2.16 : Visualisation des lignes de courants à $Re = 100$	49
Figure 3.1 : Profil de Wortmann FXS02-196.....	53
Figure 3.2 : Machine de découpe numérique par fil chaud.....	55
Figure 3.3 : Indicateur de vitesse.....	56
Figure 3.4 : Balance pour le calcul des forces de portance et de traîné.....	56
Figure 3.5 : Instrument de mesure des pressions.....	57
Figure 3.6 : Soufflerie subsonique Armfield C2.....	58
Figure 3.7 : C_z Et C_x en fonction de l'angle d'attaque α	59
Figure 3.8 : Procédures de la simulation.....	60
Figure 3.9 : Schémas du problème.....	61
Figure 3.10 : Structure du cas FXS02196 sur OpenFOAM.....	62
Figure 3.11 : La procédure suivie dans ce cas d'étude.....	63
Figure 3.12 : Visualisation du maillage à l'angle « 0 ».....	64
Figure 3.13 : C_z en fonction de l'angle d'attaque.....	67
Figure 3.14 : C_x en fonction de l'angle d'attaque.....	67
Figure 3.15 : Visualisation du champ de vitesse pour des angles d'attaque différents.....	68
Figure 3.16 : Comparaison de C_z et C_x en fonction de l'angle d'attaque.....	69
Figure 3.17 : Profil NACA4412.....	70
Figure 3.18 : Schémas du problème.....	70
Figure 3.19 : Structure du cas NACA4412 sur OpenFOAM.....	71
Figure 3.20 : Visualisation du maillage a l'incidence « 0 ».....	72
Figure 3.21 : Comparaison de C_z en fonction d'angle d'attaque.....	75
Figure 3.22 : Comparaison de C_x en fonction d'angle d'attaque.....	75
Figure 3.23 : Comparaison de la polaire.....	76
Figure 3.24 : Evolution de C_p en fonction de la corde pour $\alpha = -6^\circ, -4^\circ$ et -2°	77

Figure 3.25 : Evolution de C_p en fonction de la corde pour $\alpha = 0^\circ, 2^\circ$ et 4°	77
Figure 3.26 : Evolution de C_p en fonction de la corde pour $\alpha = 8^\circ, 12^\circ$ et 16°	79
Figure 3.27 : Evolution de C_p en fonction de la corde pour $\alpha = 18^\circ$ et 20°	80
Figure 3.28 : Visualisation du champ de vitesse a différent angle d'attaque.....	81
Figure 4.1 : Le drone Skywalker.....	83
Figure 4.2 : Modélisation de l'avion Skywalker avec Solidworks.....	85
Figure 4.3 : Diagramme des procédures suivis.....	89
Figure 4.4 : Représentation du domaine de calcule et la zone de raffinement....	89
Figure 4.5 : Visualisation du maillage obtenu par snappyHexMesh.....	93
Figure 4.6 : Visualisation du maillage dynamique.....	98
Figure 4.7 : Comparaison de la polaire du drone Skywalker.....	100
Figure 4.8 : Visualisation du champ de vitesse.....	104
Figure 4.9 : Visualisation des lignes de courants autour de l'avion.....	108
Figure 5.1 : Différentes géométries de tuyère supersonique.....	110
Figure 5.2 : La tuyère supersonique.....	111
Figure 5.3 : Schéma de la tuyère supersonique.....	112
Figure 5.4 : Régime de sur-détente en 1D.....	113
Figure 5.5 : Les régimes d'écoulement dans une tuyère supersonique. Evolution de la pression le long de la tuyère.....	114
Figure 5.6 : Schémas du problème.....	115
Figure 5.7 : Maillage d'une tuyère carré par blockMesh.....	118
Figure 5.8 : Maillage d'une tuyère circulaire par blockMesh.....	118
Figure 5.9 : Maillage d'une tuyère circulaire non axis-symétrique par blockMesh	118
Figure 5.10 : Maillage d'une tuyère elliptique par blockMesh.....	118
Figure 5.11 : Maillage d'une tuyère rectangulaire par blockMesh.....	118
Figure 5.12 : Maillage d'une tuyère triangulaire par blockMesh.....	119
Figure 5.13 : Champs des iso-Mach dans la tuyère axisymétrique.....	121
Figure 5.14 : Champs des iso-bar dans la tuyère axisymétrique.....	121
Figure 5.15 : Champs des isothermes dans la tuyère axisymétrique.....	121
Figure 5.16 : Courbe des évolutions des paramètres thermodynamiques pour la tuyère axisymétrique.....	122
Figure 5.17 : Champs des iso-Mach dans la tuyère carrée.....	123
Figure 5.18 : Champs des isothermes dans la tuyère carrée.....	123

Figure 5.19 : Champs des isobares dans la tuyère carrée.....	123
Figure 5.20 : Champs des iso-Mach dans la tuyère circulaire non axisymétrique.	124
Figure 5.21 : Champs des isobares dans la tuyère circulaire non axisymétrique..	124
Figure 5.22 : Champs des isothermes dans la tuyère circulaire non axisymétrique.....	124
Figure 5.23 : Champs des iso-Mach dans la tuyère circulaire elliptique.....	125
Figure 5.24 : Champs des isobares dans la tuyère circulaire elliptique.....	125
Figure 5.25 : Champs des isothermes dans la tuyère elliptique.....	125
Figure 5.26 : Champs des iso-Mach dans la tuyère rectangulaire.....	126
Figure 5.27 : Champs des isobares dans la tuyère rectangulaire.....	126
Figure 5.28 : Champs des isothermes dans la tuyère rectangulaire.....	126
Figure 5.29 : Champs des iso-Mach dans la tuyère triangulaire.....	127
Figure 5.30 : Champs des isothermes dans la tuyère triangulaire.....	127
Figure 5.31 : Champs des isobares dans la tuyère triangulaire.....	127
Figure 5.32 : Champs des iso-Mach dans la tuyère circulaire.....	128
Figure 5.33 : Champs des iso-Mach dans la tuyère carrée.....	128
Figure 5.34 : Champs des iso-Mach dans la tuyère circulaire non axisymétrique.	128
Figure 5.35 : Profil des iso-Mach dans la tuyère elliptique.....	129
Figure 5.36 : Profil des iso-Mach dans la tuyère carrée.....	129
Figure 5.37 : Profil des iso-Mach dans la tuyère triangulaire.....	129

LISTE DES TABLEAUX

Tableau 1.1 : Les commandes de convertissions de maillage.....	20
Tableau 1.2 : Les solveurs disponibles sur OpenFOAM.....	22
Tableau 1.3 : Informations sur le système utilisé.....	27
Tableau 2.1 : Conditions aux limites.....	45
Tableau 3.1 : Caractéristique de la soufflerie.....	57
Tableau 3.2 : Résultat expérimental.....	59
Tableau 3.3 : Conditions aux limites.....	65
Tableau 3.4 : Conditions aux limites.....	74
Tableau 4.1 : Conditions aux limites.....	99
Tableau 5.1 : Conditions aux limites	120

ANNEXES

A1. Téléchargement et installation d'OpenFOAM

Le téléchargement et l'installation d'OpenFOAM peut être un peu long pour les utilisateurs qui n'ont aucune notion sur Linux, mais en suivant les instructions recommander on peut facilement le faire.

A1.1. Sous Windows

Il existe plusieurs moyens pour installer OpenFOAM sous Windows. Soit sur un environnement virtuel, comme Cygwin, ou sur une machine virtuelle, par exemple avec Virtuel-Box. Cygwin simule un environnement Linux sous Windows, rendant possible l'exécution de ces logiciels après une simple compilation. Le lien suivant donne tout les instructions à suivre avec les téléchargements nécessaires:

<http://www.cfdsupport.com/install-openfoam-for-windows.html>

Une autre solution s'agit sur l'installation d'un émulateur qui permet d'installer une machine virtuel avec un autre système d'exploitation. L'utilisateur peut télécharger VirtuelBox sur le lien <https://www.virtualbox.org/wiki/Downloads> et choisir la plateforme ou installer le logiciel. Après, l'utilisateur doit choisir un CD/DVD du système d'exploitation Linux pour l'installer dans la machine virtuel. L'installation d'OpenFOAM est ensuite tout comme une installation sous Linux.

A1.2. Sous Linux

Nous avons choisi le système Ubuntu puisque il est probablement le plus célèbre. Les instructions suivantes sont dédiées pour Ubuntu 14.10.

On commence toujours par l'ouverture du terminal avec « Ctrl+Alt+t ».

```
# Pour passer en mode super utilisateur.
su -
# Installation des packages nécessaires.
apt-get install build-essential cmake flex bison zlib1g-dev qt4-dev-tools\
libqt4-dev libqtwebkit-dev gnuplot libreadline-dev libncurses5-dev\
libxt-dev libopenmpi-dev openmpi-bin libboost-system-dev\
libboost-thread-dev libgmp-dev libmpfr-dev\ python python-dev\
libglu1-mesa-dev libqt4-opengl-dev
# Sortir du mode super utilisateur.
exit
# Téléchargement et Installation d'OpenFOAM.
cd ~
mkdir OpenFOAM
cd OpenFOAM
```

```

wget "http://downloads.sourceforge.net/foam/OpenFOAM-2.3.0.tgz?use_mirror=mesh" -O OpenFOAM-2.3.0.tgz
wget "http://downloads.sourceforge.net/foam/ThirdParty-2.3.0.tgz?use_mirror=mesh" -O ThirdParty-2.3.0.tgz
tar -xzf OpenFOAM-2.3.0.tgz
tar -xzf ThirdParty-2.3.0.tgz
ln -s /usr/bin/mpicc.openmpi OpenFOAM-2.3.0/bin/mpicc
ln -s /usr/bin/mpirun.openmpi OpenFOAM-2.3.0/bin/mpirun
# La suite dépend sur l'architecture du système installer.
uname -m
# Si le résultat est
# Pour i686
    #Le nombre de processeur ici est 4.
    source $HOME/OpenFOAM/OpenFOAM-2.3.0/etc/bashrc
    WM_NCOMPPROCS=4 WM_MPLIB=SYSTEMOPENMPI
    WM_ARCH_OPTION=32
# Pour x86_64
    #Le nombre de processeur ici est 4.
    source $HOME/OpenFOAM/OpenFOAM-2.3.0/etc/bashrc
    WM_NCOMPPROCS=4 WM_MPLIB=SYSTEMOPENMPI

echo "alias of230='source \${HOME}/OpenFOAM/OpenFOAM-2.3.0/etc/bashrc
\$FOAM_SETTINGS'">> $HOME/.bashrc
cd $WM_THIRD_PARTY_DIR
export QT_SELECT=qt4
# Cette commande peut prendre plusieurs minutes.
./Allwmake > make.log 2>&1
#installation de Paraview
wmSET $FOAM_SETTINGS
    export QT_SELECT=qt4
    sed -i -e
    's=MPI_ARCH_PATH/include=MPI_ARCH_PATH/include;$MPI_INCLUDE='
    etc/tools/ParaView4Functions
    sed -i -e 's=//#define GLX_GLXEXT_LEGACY=#define
    GLX_GLXEXT_LEGACY='\
    ParaView-4.1.0/VTK/Rendering/OpenGL\
    vtkXOpenGLRenderWindow.cxx
    cd $WM_THIRD_PARTY_DIR/ParaView-4.1.0

wget
http://www.paraview.org/pipermail/paraview/attachments/20140210/464496cc/attach
ment.bin -O Fix.patch
    patch -p1 < Fix.patch
    cd VTK
    wget
    https://github.com/gladk/VTK/commit/ef22d3d69421581b33bc0cd94b647da7
    3b61ba96.patch -O Fix2.patch
    patch -p1 < Fix2.patch

```

```

cd ../..
uname -m

# Pour i686
# Sa peut prendre jusqu'à 2 heures.
./makeParaView4 -python -mpi -python-lib /usr/lib/i386-linux-
gnu/libpython2.7.so.1.0 > log.makePV 2>&1

# Pour x86_64
./makeParaView4 -python -mpi -python-lib /usr/lib/x86_64-linux-
gnu/libpython2.7.so.1.0 > log.makePV 2>&1
wmSET $FOAM_SETTINGS
# Compilation d'OpenFOAM.
# Sa peut prendre quelques heures jusqu'à 6 heures.
cd $WM_PROJECT_DIR
export QT_SELECT=qt4
./Allwmake > make.log 2>&1
# On vérifier si l'installation est correctement terminer en lançant un tutoriel.
of230
tut
cd basic/potentialFoam/cylinder
./Allrun

```

A2. Le Script « Exécution »

```

#!/bin/bash
# Les différents angles alfa
for alfa in -6 -4 -2 0 1 2 3 4 5 6 7 8 10 12 14 15 16 18 20
do
cd alpha$alfa # Entrer dans le répertoire alfa.
# Parametres du fichier foilgmsh.m.
archi="NACA4412.DAT"
yplus=1
eter="a"
Re=3000000
M=0.004205206
T0=300
N="[1000,800,800,800]"
bump=1
octave --silent --eval "foilgmsh
('$archi','$alfa','$yplus','$eter','$Re','$M','$T0','$N','$bump)"
gmsh -3 NACA4412.DAT.geo # Constitution du maillage par Gmsh.
gmshToFoam NACA4412.DAT.msh # Convertissions du maillage.
decomposePar # Décomposition du maillage par processeur.
# Lancement du solveur simpleFoam en calcul parallèle.
mpirun -np 2 simpleFoam -parallel > simpleFoam.log 2>&1
reconstructPar # Reconstruction du maillage.
sample # Calcul des coefficients de pression.

```

```

sed -i 1,3d postProcessing/surfaces/2500/airfoil/scalarField/p
perl -pi -e "s/\\//g" postProcessing/surfaces/2500/airfoil/scalarField/p
gfortran Cp.f90      # Compiler le programme Cp.f90.
./a.out             # Exécution du programme Cp.f90.
./Cp.plot           # Dessiner la courbe de coefficient de pression.
# Suppression des fichiers inutiles.
rm -r NACA4412.DAT.geo NACA4412.DAT.msh simpleFoam.log processor0
processor1
cd .. # Sortir du dossier alfa
done # Changer l'angle et refaire les calculs
# Collection des résultats de chaque angle d'attaque.
# Different angles alfa
for alfa in -6 -4 -2 0 1 2 3 4 5 6 7 8 10 12 14 15 16 18 20
do
tail alpha$alfa/postProcessing/forceCoeffs/0/forceCoeffs.dat -n 1 | cut -d \" -f 6 >
alpha$alfa
done
grep "000e+0" alfa-6 alfa-4 alfa-2 alfa0 alfa1 alfa2 alfa3 alfa4 alfa5 alfa6 alfa7 alfa8
alfa10 alfa12 alfa14 alfa15 alfa16 alfa18 alfa20 | cut -d \" -f 6 > sample
rm alpha* # Supprimer les fichiers inutiles
perl -pi -e 's/alfa//g' sample
perl -pi -e 's:/\//g' sample
./plot # Dessiner les courbes des coefficient de traîné et de portance.

```

A3. Visualisation des lignes de courants sous Paraview

A la fin des calculs, la phase de post-traitement peut être lancée avec Paraview par la commande paraFoam. Ce logiciel de post-traitement contient plusieurs options de visualisation. On a choisi de montrer a titre d'exemple comment voir les lignes de courants autour de l'avion Skywalker sous Paraview.

```
$ paraFoam
```

Dans la fenetre « Propriétés » on clique sur « Apply ». Ensuite on tape sur « **Ctrl+Espace** » et on écrit : « Stream tracer » puis « **Entrer** ».

Dans la fenêtre Propriétés, et dans la partie « Number of points » on choisi 500. Pour mieux voir les lignes on peut augmenter changer « Line Width » par 1,5 et l'effet de transparence dans « Opacity » jusqu'à 0.8. Enfin on va sur le bouton Open et on choisi notre fichier « *.OpenFOAM » mais cette fois avant de cliquer sur « Apply » on choisi dans « Mesh Parts » uniquement le groupe « wall ».