

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ BATNA 2
FACULTE DE TECHNOLOGIE



MEMOIRE DE FIN D'ETUDES

PRESENTE POUR OBTENIR LE DIPLOME DE

MASTER

Spécialité : *ENERGITIQUE*

PAR : LAOUICHI SIF ELHAK

THEME

Simulation numérique bidimensionnelle autour d'un modèle de véhicule

Soutenue le 20/06/2016

Proposé et dirigé par: **Dr. Laïd MESSAOUDI**

Année Universitaire 2015-2016

DÉDICACE

Je dédie ce travail

à mes parents qui j'espère qu'ils

Trouveront ici le fruit de leur amour et leur souffrance pour moi.

A mon père et ma mère.

A mes frères et tous les amis.

*A toute ma grande famille ainsi que tous mes amis et camarades
d'étude spécialement Badro, Rahim et Salah.*

A tous ceux qui me connaissent et qui ne me connaissent Pas de près et de loin.

Je dédie chaleureusement ce travail.

S. LAOUICHI.

REMERCIEMENTS

*Je tiens à remercier du fond du coeur **Dr. Laïd MESSAOUDI** Maître de Conférences à l'université Batna 2, Département de Génie Mécanique, de m'avoir proposé ce sujet et de m'avoir encadré. Je le remercie vivement pour tous ses conseils ainsi que de sa grande disponibilité.*

Je tiens aussi à remercier, à l'avance, les membres du jury qui seront désignés pour évaluer ce travail.

Je remercie également tous les enseignants du département de Génie Mécanique qui ont contribué à ma formation.

Enfin, je remercie tous mes amis qui m'ont soutenus et tous ceux qui m'ont aidés de près ou de loins pour aboutir à ce modeste travail.

S. LAOUICHI.

TABLE DES MATIÈRES

Liste des figures	iii
Liste des tableaux	v
1 Notion général sur la CFD	4
1.1 Introduction	4
1.2 Notion de maillage	4
1.2.1 Composants du maillage	4
1.2.2 Types de maillage	5
1.2.3 Techniques générales de génération du maillage	7
1.3 Modèles de turbulence	8
1.4 Logiciels de CFD	12
1.5 Conclusion	13
2 Conception et maillage	14
2.1 Introduction	14
2.2 Description des l'expériences [4]	14
2.3 Description de Gmsh	15
2.3.1 Géométrie dans Gmsh	16
2.3.2 Maillage avec Gmsh	16
2.4 Création de la géométrie	16
2.5 Maillage du modèle	17
2.5.1 Maillages non structurés	18
2.5.1.1 Maillage avec un champ rectangulaire "Box field"	18
2.5.1.2 Maillage avec un champ circulaire "Ball field"	19
2.5.2 Maillage structuré "Multi-Bloc"	19
2.5.3 Maillage Hybride	21

2.6	Conclusion	21
3	Simulations et résultats	23
3.1	Introduction	23
3.2	Présentation d'OpenFOAM	23
3.2.1	Solveurs dans OpenFoam	24
3.2.2	Dossier de travail	24
3.2.3	Conversion lu maillage	25
3.3	Influence du types de maillage	26
3.3.1	Simulations	26
3.3.2	Résultats	27
3.3.2.1	Répartition de vitesse	29
3.3.2.2	Répartition de pression	31
3.3.2.3	Variation de k	32
3.3.2.4	Comparaison	32
3.4	Influence des modèles de turbulence	36
3.4.1	Simulations	36
3.4.2	Résultats	37
3.4.2.1	Répartition de vitesse	37
3.4.2.2	Répartition de pression	38
3.4.2.3	Variation de k	38
3.4.3	Comparaison	38
3.5	Conclusion	42
	Bibliographie	44
	A Création de la géométrie avec Gmsh	47
	B Détails des maillages avec Gmsh	52
B.1	Maillage avec un champ parallélépipédique "Box field"	52
B.2	Maillage avec un champ circulaire "Ball field"	57
B.3	Maillage structuré "Multi-Bloc"	59
B.4	Maillage Hybride	61
	C Détails des simulations avec OpenFoam	63

LISTE DES FIGURES

1.1	<i>Illustration de la notion de maillage.</i>	5
1.2	<i>Quelques types d'éléments.</i>	5
1.3	<i>Exemples de maillages.</i>	6
1.4	<i>Exemples de grilles multiblocs conforme (a) et non-conforme (b).</i>	7
1.5	<i>Maillage de type hybride.</i>	7
1.6	<i>Evolution de la taille des éléments.</i>	8
2.1	<i>Vue de côté de la soufflerie avec le modèle de véhicule.</i>	15
2.2	<i>Fenêtre principale de Gmsh.</i>	15
2.3	<i>Géométrie du modèle de véhicule [? ?].</i>	17
2.4	<i>Domaine autour du modèle de voiture.</i>	17
2.5	<i>Maillage Box field.</i>	18
2.6	<i>Maillage Ball field.</i>	19
2.7	<i>Maillage multi-bloc.</i>	20
2.8	<i>Maillage hybride</i>	21
3.1	<i>Structure du dossier de travail sous OpenFoam.</i>	24
3.2	<i>Positions des stations de mesures expérimentales ($\frac{x}{L}$).</i>	26
3.3	<i>Types de conditions aux limites.</i>	27
3.4	<i>Evolution de la composant axiale de la vitesse en fonction du nombre d'itérations.</i>	28
3.5	<i>Evolution du temps de calcul.</i>	28
3.6	<i>Répartition de la vitesse autour du véhicule.</i>	29
3.7	<i>Répartition de vitesse derrière le véhicule.</i>	30
3.8	<i>Répartition de vitesse devant le véhicule.</i>	30
3.9	<i>Lignes de courant autour du véhicule.</i>	31
3.10	<i>Répartition de pression autour du véhicule.</i>	31
3.11	<i>Variation de l'énergie cinétique turbulente derrière le véhicule.</i>	32

3.12	<i>Comparaison des vitesses U_x des différents maillages dans différentes stations.</i>	33
3.13	<i>Comparaison des vitesses U_x des différents maillages dans différentes stations (suite).</i>	34
3.14	<i>Comparaison des vitesses U_y des différents maillages dans différentes stations.</i>	35
3.15	<i>Comparaison des vitesses U_y des différents maillages dans différentes stations (suite).</i>	36
3.16	<i>Répartition de vitesse autour du véhicule.</i>	37
3.17	<i>Lignes de courant autour du véhicule.</i>	37
3.18	<i>Répartition de pression autour du véhicule.</i>	38
3.19	<i>Variation de l'énergie cinétique turbulente derrière la véhicule</i>	38
3.20	<i>Comparaison des vitesses U_x des différents modèles de turbulence dans différentes stations.</i>	39
3.21	<i>Comparaison des vitesses U_x des différents modèles de turbulence dans différentes stations (suite).</i>	40
3.22	<i>Comparaison des vitesses U_y des différents modèles de turbulence dans différentes stations.</i>	41
3.23	<i>Comparaison des vitesses U_y des différents modèles de turbulence dans différentes stations.</i>	42
A.1	<i>Création d'un fichier de 'geo'.</i>	47
A.3	<i>Ouvrir le fichier de géométrie.</i>	48
A.2	<i>Recharger et éditer la géométrie.</i>	48
A.4	<i>Création des coordonnées du véhicule et les limites du domaine de calcul.</i>	49
A.5	<i>Bords et limites du véhicule et du domaine de calcul.</i>	50
A.6	<i>Création de la surface.</i>	51
B.1	<i>Extrusion de la géométrie.</i>	54
B.2	<i>Création des entités physiques.</i>	56
B.3	<i>Visualisation du maillage généré sur un domaine rectangulaire avec transfinite (à gauche) et avec transfinite et recombine (à droite).</i>	59
B.4	<i>Décomposition Multibloc du domaine de calcul.</i>	60
B.5	<i>Fenêtre de transformation.</i>	62
C.1	<i>Structure du dossier de travail.</i>	63
C.2	<i>Répertoire de travail après calculs.</i>	68

LISTE DES TABLEAUX

1.1	<i>Qualité associée au facteur de distorsion.</i>	8
1.2	<i>Modèles statistiques de turbulence de type RANS.</i>	11
3.1	<i>Coefficient de traînée.</i>	29

INTRODUCTION GÉNÉRALE

L'étude de l'écoulement autour des véhicules peut être réalisée avec des expériences dans les souffleries ou par simulations numériques. Les souffleries ont été largement utilisées pendant des années dans l'aérodynamique automobile afin d'améliorer la conception des véhicules modernes [7],[9]. Dans le passé, ces expériences consistaient principalement à la mesure des coefficients aérodynamiques et visualisation de l'écoulement autour des véhicules. Bearman [6] a réalisé des expériences dans une soufflerie pour mesurer le sillage des modèles de véhicules. Rauser et al. [15] ont réalisés des expériences dans une soufflerie avec l'utilisant des structures d'un véhicule simple à l'échelle 1/5. Ils ont étudié les effets de la conception du soubassement des véhicules aérodynamiques et on trouvé des solutions pour les appliquer sur des productions en série. Leurs expériences ont montré qu'il est possible d'améliorer encore l'aérodynamique des organes de type de production en ajoutant un habillage soubassement efficace.

Ces configurations ont déjà été utilisées pour les voitures de course, mais pas aux véhicules pour une utilisation quotidienne. Ahmed [1],[2] a réalisé une série d'expériences en soufflerie afin d'examiner la structure de sillage d'un modèle d'automobile réaliste. Ahmed et al.[3] ont également réalisé des expériences sur une modèle de véhicule et ont présenté les mesures de pression du sillage et mesuré les forces pour différents angles d'inclinaison de la lunette arrière. Ils ont montré que près de 85% de la traînée du corps est une traînée de pression et la plupart de cette traînée est générée à l'extrémité arrière. Ils ont également montré que l'écoulement de sillage présente un système à trois étages de tourbillons en fer à cheval où la force et l'existence de celles-ci dépendent de l'angle d'inclinaison.

Pendant ces dernières années, des progrès significatifs ont été enregistrés dans le domaine de la CFD "*Computational Fluid Dynamics*" (« Dynamique des Fluides Numérique ») qui est progressivement devenue un outil essentiel dans la conception et le design des véhicules. La CFD, est un ensemble de méthodes numériques permettant

d'obtenir une solution approximative d'un problème de dynamique des fluides et/ou de transfert thermique. Les équations qui interviennent sont celles de la mécanique des fluides, résolues par des méthodes numériques. La solution est approximative et non pas exacte pour plusieurs raisons. D'abord, parce qu'on résout les équations de Navier-Stokes numériquement en les discrétisant. Deuxièmement, et comme on le verra un peu plus loin, pour des raisons de limitation de la puissance de calcul et de la mémoire, certains termes des équations à résoudre sont remplacés par des modèles empiriques qui ne sont pas exacts ; c'est en particulier le cas lorsque les écoulements à modéliser sont en régimes turbulents. Cependant, grâce au développement des méthodes numériques et à des calculateurs de plus en plus puissants avec une grande capacité de mémoire, la CFD permet d'avoir des solutions très satisfaisantes. Ceci est encore plus vrai dans la plupart des domaines de l'industrie où très souvent une prédiction de l'ordre de grandeurs des valeurs moyennes est amplement suffisante.

L'utilisateur typique d'ordinateurs possède des quantités de logiciels qu'il a acquis de par le passé et qu'il n'utilise plus désormais. Il a peut-être acheté un ordinateur plus performant ou changé de constructeur, ce qui a empêché ses programmes de continuer à fonctionner. Les logiciels sont peut-être devenus obsolètes, ou ont cessé de répondre à ses besoins. Il a peut-être acheté deux ordinateurs ou plus, et ne souhaite pas déboursier quoi que ce soit pour obtenir une deuxième copie du même logiciel. Quelle qu'en soit la raison, le logiciel qui lui a coûté de l'argent voici quelques années ne remplit plus son rôle. Cette situation est-elle inévitable ? Si tel était le cas, vous continueriez sans doute à utiliser les logiciels que vous avez achetés il y a des années. Ce sont là quelques-uns des droits que l'Open Source (OS) vous garantit. La définition de l'Open Source est une déclaration des droits de l'utilisateur d'ordinateur. Elle définit certains droits que la licence d'un logiciel doit vous garantir pour pouvoir être qualifiée d'Open Source. Ceux qui ne proposent pas des programmes OS trouvent que la concurrence est de plus en plus rude avec ceux qui ont compris que les utilisateurs commencent à apprécier et à exiger les droits qu'ils auraient dû avoir depuis toujours. Des programmes comme ceux qui constituent le système d'exploitation Gnu/Linux et le navigateur web de Netscape ont acquis une grande popularité, en prenant la place de logiciels couverts par des licences plus restrictives. Les sociétés qui utilisent du logiciel OS profitent des avantages liés à un modèle de développement très réactif, souvent mis en place par plusieurs sociétés qui coopèrent, et dont une grande partie du travail est fournie par des individuels qui ont tout simplement besoin d'une amélioration qui réponde mieux à leurs besoins. Les volontaires qui ont créé des produits comme Linux n'existent, et les sociétés ne peuvent coopérer, que grâce aux droits fournis par l'Open Source. Le programmeur moyen se sentirait lésé s'il investissait énormément de travail dans un programme, pour constater que le propriétaire de ce dernier se contente de vendre la

version améliorée sans rien lui proposer en retour. Mais ces mêmes programmeurs n'ont pas peur d'apporter des contributions à l'OS, car ils savent que les droits suivants leur sont alors garantis : le droit de faire des copies du programme, et de distribuer ces copies, le droit d'accéder au code source du logiciel, un préliminaire nécessaire pour pouvoir y apporter des modifications, le droit d'améliorer le programme. Ces droits comptent pour celui qui contribue à améliorer des logiciels car ils maintiennent tous les développeurs au même niveau. Quiconque le souhaite a le droit de vendre un programme OS, aussi les prix seront-ils bas et le développement destiné à conquérir de nouveaux marchés sera-t-il rapide. Quiconque investit de son temps à construire des connaissances sous la forme d'un programme OS peut fournir de l'aide sur ce dernier, et cela fournit aux utilisateurs la possibilité de mettre en place leur propre assistance technique, ou de choisir parmi des assistances techniques en concurrence les unes avec les autres. Tout programmeur peut spécialiser un programme OS pour le rendre utilisable dans des marchés spécifiques afin d'atteindre de nouveaux consommateurs. Ceux qui se lancent dans de telles entreprises ne sont pas tenus d'acquitter des royalties ou la moindre licence. La raison de la réussite de cette stratégie vient du fait que l'économie de l'information est fondamentalement différente de l'économie qui règle la consommation des autres produits. On peut copier une information telle qu'un programme d'ordinateur pour un prix dérisoire. L'électricité qu'il en coûte ne dépasse par l'euro-cent, pas plus que l'utilisation de l'équipement nécessaire à cet acte. En comparaison, on ne peut pas recopier une miche de pain sans dépenser une livre de farine[13].

L'objectif principal de cette étude est l'utilisation de logiciels OpenSource sous Linux, une alternative aux logiciels commerciaux sous Windows. Cette étude devra montrer les capacités des outils OpenSource à modéliser, mailler, simuler numériquement et présenter les résultats (Pré-processing, Solvers, Post-processing). Aussi, les outils de présentation et de rédaction devront être issus de la plateforme Linux. L'exemple considéré est un modèle 2D de véhicule dont nous disposons des coordonnées ainsi que les résultats expérimentaux. Il s'agit dans cette étude de voir l'influence de différents types de maillages et de modèles de turbulence sur l'écoulement autour de ce modèle.

CHAPITRE 1

NOTION GÉNÉRAL SUR LA CFD

1.1 Introduction

Dans ce chapitre, nous allons passer en revue quelques notions élémentaires et essentielles sur les étapes de la CFD. Une particularité sera donnée à l'étape du maillage qui constitue la clé de toute simulation numérique.

1.2 Notion de maillage

Toute simulation CFD commence par la réalisation de la géométrie en 2D ou 3D, soit avec un logiciel intégré au code de calcul CFD soit à l'aide d'un logiciel de CAO. Dans le deuxième cas, la géométrie doit être exportée en un format lisible par le logiciel de maillage.

La génération du maillage (2D ou 3D) est une phase très importante dans une analyse CFD, vu son influence sur la solution calculée. Un maillage de très bonne qualité est essentiel pour l'obtention d'un résultat de calcul précis, robuste et signifiant. La qualité du maillage a un sérieux impact sur la convergence, la précision de la solution et surtout sur le temps de calcul. Une bonne qualité de maillage repose sur la minimisation des éléments présentant des « distorsions » (*'skewness'*), et sur une bonne « résolution » dans les régions présentant un fort gradient (couches limites, ondes de choc, . . .). Un bon maillage doit également être suffisamment « lisse ».

1.2.1 Composants du maillage

Noeuds et éléments La modélisation numérique repose sur la reformulation des équations de conservation sur des volumes élémentaires ou discrets, appelés éléments

ou mailles. Associer ces éléments, nous retrouvons les noeuds de discrétisation, c'est-à-dire les points de résolution des équations discrètes. Ceux-ci peuvent être aussi bien placés aux sommets des éléments qu'en leur centre ou encore sur les faces, selon la méthode de discrétisation utilisée (Figure 1.1)

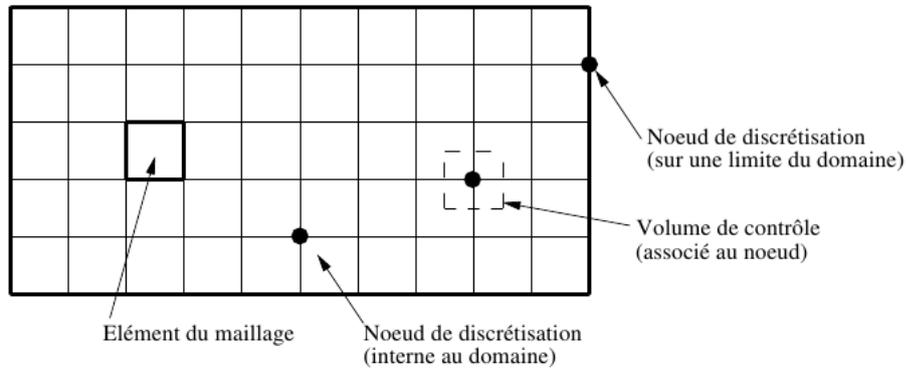


Figure 1.1: *Illustration de la notion de maillage.*

Géométrie et topologie Nous faisons la différence entre la géométrie qui caractérise la forme du domaine et la topologie qui est le résultat du découpage spatial du domaine sur lequel s'appuie le maillage. La topologie est donc une classification des objets de type segments, faces, ...etc. Nous distinguons plusieurs types de maillages, définis par le nombre de noeuds associés à chaque élément (Figure 1.2) et par le nombre de liaisons pour chaque noeuds.

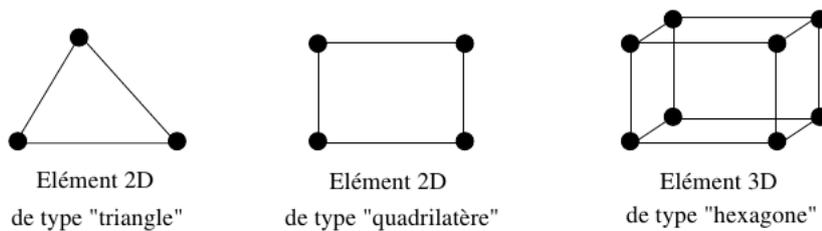
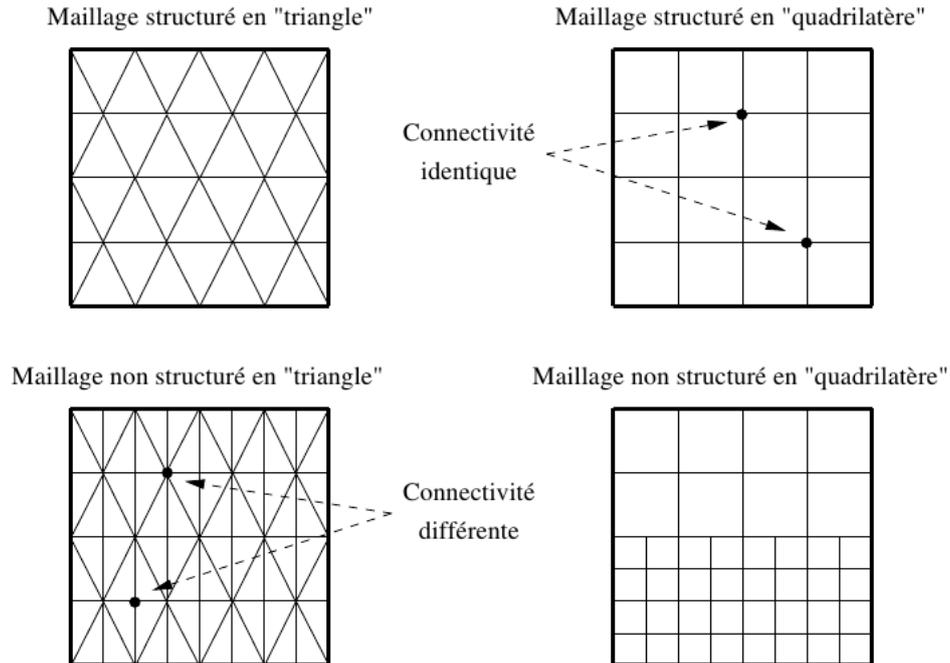


Figure 1.2: *Quelques types d'éléments.*

1.2.2 Types de maillage

Les types de maillage sont liés à la connectivité qui décrit les liaisons entre les sommets des éléments. On dit maillage structuré si les noeuds de même type (dans le domaine, sur une limite ou sur un coin) ont toujours le même nombre de noeuds voisins, ou sont associés au même nombre d'éléments. La connectivité associée à ces noeuds est alors toujours de même type. Dans le cas d'un maillage non-structuré, la connectivité est de type quelconque, et le nombre de voisins de chaque noeud diffère localement (Figure 1.3).

Figure 1.3: *Exemples de maillages.*

Le principal avantage des maillages structurés est une connaissance complète et immédiate du voisinage de chaque point de discrétisation. En effet, le nombre de noeuds est constant dans chaque direction de maillage. Cet avantage se trouve être aussi son principal inconvénient car les maillages structurés ne sont pas adaptés à tous les types de géométries.

Monobloc et multibloc : Il existe de nombreux codes industriels de génération automatique de maillage à partir de topologies plus ou moins complexes. Dans la majorité des processus industriels, les géométries utilisés sont complexes et leurs traitements génèrent de nombreuses difficultés, à la fois techniques et numériques. La mise en place du maillage est parfois délicate et peut conduire à une résolution insuffisante ou à une qualité de maillage médiocre. Lorsque la géométrie est représenté par une grille unique, le terme de maillage monobloc est utilisé. Dans le cas contraire, on parle de maillage multibloc, composé alors de plusieurs grilles monoblocs.

Conforme et non-conforme : La définition de la conformité d'un maillage multibloc est plus complexe à appréhender. Nous adopterons la définition suivante : « un maillage est dit conforme si quelle que soit la ligne de maillage, elle est continue au passage de l'interface entre les blocs » (Figure 1.4(a)). Dans ce cas, si il n'y a pas de recouvrement, chaque noeud situé sur une interface appartient aux différents blocs la constituant. La connectique d'un noeud de l'interface est alors de type structuré. Un maillage non-conforme sera donc un maillage dont les lignes sont interrompues à

l'interface (Figure 1.4(b)). Les noeuds situés aux interfaces non-conformes conduisent à un maillage localement non-structuré. Cette configuration est couramment appelée maillage structuré par bloc.

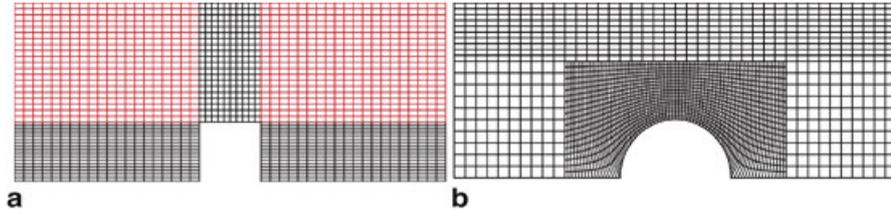


Figure 1.4: Exemples de grilles multiblocs conforme (a) et non-conforme (b).

Maillage hybride : C'est un maillage généré par un mélange d'éléments de différents types, triangulaires ou quadri-latéraux en 2D, tétraédriques, prismatiques, ou pyramidaux en 3D. Il combine les avantages des maillages structurés et non structurés. Ses avantages sont combinés entre les avantages du maillage structuré et ceux du maillage non structuré (Figure 1.5).

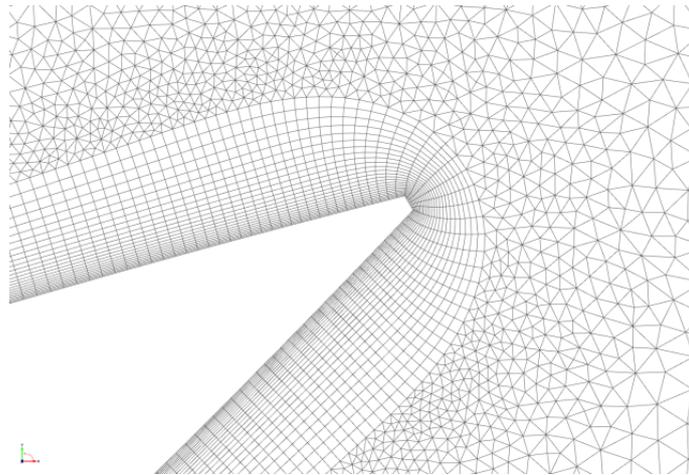


Figure 1.5: Maillage de type hybride.

1.2.3 Techniques générales de génération du maillage

En pratique, il n'existe pas de règle précise pour la création d'un maillage valable, cependant il existe différentes approches qui permettent d'obtenir une grille acceptable. Nous pouvons résumer ces règles ainsi :

- Maintenir une bonne qualité des éléments.
- Assurer une bonne résolution dans les régions à fort gradient.
- Assurer un bon lissage dans les zones de transition entre les parties à maillage fin et les parties à maillage grossier.
- Minimiser le nombre total des éléments (temps de calcul raisonnable).

Qualité d'un maillage : La génération d'une très bonne qualité de maillage est essentielle pour l'obtention d'un résultat de calcul précis, robuste et significatif. Aussi, ce maillage doit aboutir à un temps de calcul le plus faible possible.

Nombre total d'éléments : Un nombre important d'éléments de maillage permet sans doute d'améliorer la précision des calculs, mais pénalise les ressources informatiques en terme de mémoire et alourdit le système.

Distorsion : Le facteur de distorsion F_d ('skewness') peut être défini de deux façons différentes. Pour des éléments triangulaires ou tétraédriques, le calcul est basé sur le volume équilatéral. L'autre formulation, applicable pour tout élément, est basée sur la déviation angulaire.

Le facteur de distorsion est nul pour des éléments « parfaits » (carrés, triangles équilatéraux). Notons que les grandes valeurs du facteur de distorsion induisent des erreurs de calcul et ralentissent considérablement le processus de convergence. Quelques distorsions peuvent être tolérées si elles sont situées dans des régions à faible gradient. Le tableau 1.1 illustre la variation de la qualité des éléments de maillage en fonction de la valeur du coefficient de distorsion F_d :

F_d	0 – 0.25	0.25 – 0.50	0.50 – 0.80	0.80 – 0.95	0.95 – 0.99	0.99 – 1.00
Qualité	Excellente	Bonne	Acceptable	Pauvre	Très pauvre	Mauvaise

Tableau 1.1: *Qualité associée au facteur de distorsion.*

Lissage : Le changement dans la taille des éléments du maillage d'une zone maillée à une autre doit être graduel, la variation de la taille des éléments de deux zones adjacentes ne doit idéalement pas dépasser 20% à 30% (Figure 1.6).

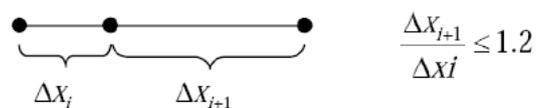


Figure 1.6: *Evolution de la taille des éléments.*

1.3 Modèles de turbulence

La modélisation de la turbulence est une question clé dans la plupart des simulations CFD. Pratiquement toutes les applications d'ingénierie sont turbulentes et ont donc besoin d'un modèle de turbulence.

Les avancées technologiques réalisées dans le domaine du calcul haute performance permettent d'avoir recours à des modèles de turbulence réduisant au minimum le degré de modélisation qu'ils contiennent, voire même s'affranchissant de leur utilisation. Ces approches sont la simulation numérique directe (DNS, '*Direct Numerical Simulation*') et la simulation des grandes échelles (LES, '*Large Eddy Simulation*'), constituent des outils de simulation numérique très puissants. Néanmoins, ces approches nécessitent encore à l'heure actuelle énormément de ressources informatiques et sont limitées à de faibles nombres de Reynolds et à des géométries simples, rendant leur usage rédhibitoire d'un point de vue industriel.

La plupart des codes de calculs industriels ont recours à l'approche aux équations de NavierStokes moyennées (RANS, '*Reynolds Averaged Navier-Stokes*') qui possède un haut degré de modélisation mais permet d'obtenir des résultats dans des délais très courts. Dans le cadre des simulations numériques réalisées pour l'aéronautique, les modèles de turbulence utilisés de façon classique en approche RANS sont très performants. Pour ces modèles, la température n'est souvent qu'un scalaire passif et dépend du champ de vitesse : le couplage entre les champs dynamique et thermique, réalisé par des termes dits de flottabilité, est ignoré. Lorsque la température joue un rôle moteur, comme c'est le cas des écoulements de convection naturelle, le couplage entre les champs dynamique et thermique est essentiel : les termes de flottabilité doivent être fidèlement représentés.

Les codes de calculs industriels utilisent le plus souvent des modèles RANS à viscosité turbulente. Toutefois, la flottabilité induit une direction privilégiée dans l'écoulement et ce phénomène ne peut être reproduit que par les modèles anisotropes, ce que ne sont pas les modèles à viscosité turbulente. Cette constatation a d'ailleurs été appuyée par Hanjalić (2002)[8] qui a montré que les modèles de turbulence ayant recours à cette hypothèse ne sont pas suffisants pour pouvoir représenter les effets de la flottabilité. L'option la plus directe est de résoudre les équations de transport des tensions de Reynolds et des flux de chaleur turbulents. Une autre option consiste à utiliser un '*Differential Reynolds Stress Model*' (DRSM) avec un modèle algébrique pour les flux de chaleur turbulents. Toutefois, ces solutions restent coûteuses du point de vue des ressources informatiques. Dans cette optique, le couplage de modèles algébriques à la fois pour les tensions de Reynolds et les flux de chaleur turbulents constitue une alternative intéressante. Ces modèles algébriques prennent en compte l'anisotropie induite par la flottabilité pour un faible coût. Les équations dynamiques et thermiques sont alors couplées grâce à la présence du terme de flottabilité.

Quelques modèles procédant au couplage de modèles algébriques ont été développés ces quinze dernières années. Toutefois, ces modèles ne sont pas idéaux et ne sont pas

envisageables au sein de codes de calcul industriels. Leur domaine d'application est restreint à un nombre de géométries limité, la représentation du couplage est souvent simplifiée ou leur formulation est compliquée à mettre en œuvre. L'utilisation de ces modèles est donc plus ou moins couronnée de succès selon les configurations choisies [17].

Modèles RANS : Nous nous limitons ici à des modèles RANS stationnaires, permettant d'obtenir une bonne approximation des valeurs moyennes dans des écoulements industriels. Les principaux modèles de type RANS [11] disponibles sont listés dans le tableau(1.2).

Modèles	Avantages	Inconvénients
Spalart-Allmaras	Economique (1 équ). Bon pour les écoulements libres, et les écoulements sur profil sans décollements ni grands gradients de pression.	Nécessite une plus grande résolution du maillage aux frontières (pas de lois aux murs).
Standard $k - \varepsilon$	Robuste, économique et relativement précis. Adapté aux écoulements à grand nombre de Reynolds.	Résultats médiocre pour des écoulements complexes (fort gradient de pression, rotation et swirl).
$RNGk - \varepsilon$	Dérivé par une méthode statistique rigoureuse (Renormalization Group theory). Bon pour des écoulements moyennement complexes (impact de jet, séparation d'écoulements, recirculations). Respecte une contrainte physique que peut violer le modèle .	Limité par l'hypothèse de viscosité turbulente isotrope.
Realizable $k - \varepsilon$	Offre les mêmes avantages que le $RNG k - \varepsilon$. Recommandé dans le cas des turbomachines.	Limité par l'hypothèse de viscosité turbulente isotrope.
SST et Standard $k - \omega$	Modèle recommandé pour les problèmes liés aux turbomachines (à comparer au Realizable $k - \varepsilon$). La version $SST k - \omega$ consiste en une transition entre le modèle $k - \omega$ standard (développé pour les nombres de Reynolds modérés et les couches limites) et une version haut Re du $k - \varepsilon$ lorsqu'on est loin des parois.	Nécessite une plus grande résolution du maillage aux frontières (pas de lois aux murs).
Reynolds-Stress Model (RSM)	Le modèle le plus complet physiquement (transport et anisotropie de la turbulence sont pris en compte).	Requiert plus de temps CPU. Les équations de quantité demouvement et de transport de la turbulence sont étroitement liées.

Tableau 1.2: Modèles statistiques de turbulence de type $RANS$.

1.4 Logiciels de CFD

Il y a beaucoup des logiciels pour résoudre des problèmes CFD commercial ou Open Source (c'est-à-dire libre, ouvert et gratuit...). Parmi ceux-ci on va présenter les plus connus :

- **OpenFOAM** : C'est un logiciel libre, open source, multiplateforme et multi-physique spécialement destiné à la Dynamique des Fluides Numérique (CFD) et développé par la société britannique *OpenCFD Ltd* mais initialement conçu à l'Imperial College de Londres dans les années 1980 par Henry Weller. *OpenFOAM* est codé en C++, utilise une approche orientée objet et est principalement dédié à la résolution des équations aux dérivées partielles par la méthode des volumes finis. Ses principaux avantages sont qu'il est gratuit, performant, parallélisable et personnalisable, c'est-à-dire que des modules complémentaires peuvent être développés car sa source est accessible[12].
- **ANSYS Fluent** : C'est un code CFD qui permet la prédiction des écoulements fluides par résolution des équations de bilan en se basant sur la méthode des volumes finis et des schémas de discrétisation existant dans sa banque de données. *Fluent* offre un très large éventail de fonctionnalités pour modéliser les phénomènes physiques d'écoulement fluide, de turbulence, de transfert de chaleur et de réactions chimiques. Ses applications industrielles vont de l'écoulement d'air autour d'une aile d'avion, à la combustion à l'intérieur d'un four; des colonnes à bulles aux plates-formes pétrolières; du flux sanguin à la fabrication de semi-conducteurs; de la conception de salles blanches aux usines de traitement d'eaux usées. Des modèles spécifiques dédiés à la combustion interne, l'aéroacoustique, les turbomachines et les systèmes multiphasiques viennent élargir la panoplie de ses applications[5].
- **ANSYS CFX** : C'est un logiciel de CFD généraliste extrêmement performant, utilisé pour résoudre une large variété de problèmes d'écoulement fluide depuis plus de 20 ans. Grâce à son solveur avancé, moderne et parallélisé, les utilisateurs obtiennent rapidement des solutions fiables et robustes. Le vaste choix de modèles physiques proposé permet de représenter la quasi-totalité des phénomènes d'écoulement fluide. Le solveur tout comme les modèles physiques, sont accessibles via un environnement utilisateur graphique moderne, intuitif, flexible, personnalisable et automatisable (fichiers de session, scripts, générateur d'expression...)[5].
- **StarCCM+** : C'est un logiciel de CFD développé par *CD-Adapco* en 2004. *StarCCM+* est une amélioration du logiciel *STAR-CD* pour un produit plus innovant est performant. L'intérêt de ce logiciel est de résoudre des problèmes de mécanique des fluides, en résolvant simultanément les problèmes de flux et de transferts de chaleur, alors que d'autres logiciels utilisent deux solveurs différents

couplés. *StarCCM+* qui intègre son propre mailleur en plus de son solveur. De plus, contrairement à d'autres simulateurs, il utilise le maillage en trois dimensions. Enfin, *StarCCM+* a été le premier logiciel à utiliser la fonction polyhedrale pour le maillage et cette fonction s'est révélée plus précise que les autres maillages proposés avant [10].

1.5 Conclusion

Dans ce chapitre, nous avons passé en revue les différents type de maillages, des notions de CFD, les modèles de turbulence utilisés et finalement les différents logiciels de grande envergure utilisés par les chercheurs et les industriels dans le monde.

Le prochain chapitre sera consacré à la conception de la géométrie du véhicule ainsi qu'à son maillage.

CHAPITRE 2

CONCEPTION ET MAILLAGE

2.1 Introduction

Notre travail est basé principalement sur un article de référence [4] qui traite l'étude expérimentale et numérique de l'écoulement turbulent bidimensionnel et stationnaire autour d'un modèle de véhicule. Le modèle de turbulence utilisé est le $k - \varepsilon$. Les perspectives de cet article recommandent l'utilisation d'autres types de maillages et de modèles de turbulence. C'est dans ce but que nous avons dirigé notre travail afin de compléter cette étude mais en utilisant que des produits OpenSource. Nous allons donc, dans ce chapitre, détailler l'utilisation de ces produits gratuits afin de résoudre ce problème numérique.

2.2 Description des l'expériences [4]

Les expériences ont été réalisées dans un soufflerie qui est à 3/4 ouverte sur les côtés et le haut (Figure2.1). La vitesse maximale de l'air peut atteindre 15 m/s et le niveau de turbulence est inférieure à 0.3%. La longueur, la hauteur et la durée de la section de test sont 2.8, 1.4 et 1,8m respectivement. Pour positionner les sondes dans le soufflerie, et pour tourner la table de la soufflerie, un système de déplacement sept axes est utilisé. Les imprécisions linéaires et le déplacement angulaire sont inférieure à 0.05 et 0.2 mm, respectivement. L'acquisition et le traitement des données se fait par un PC. Pour les mesures de LDA,¹ le logiciel commercial *Blastware* est utilisé. Un progiciel a été développé Pour les mesures à fil chaud.

1. laser-Doppler-anemometry

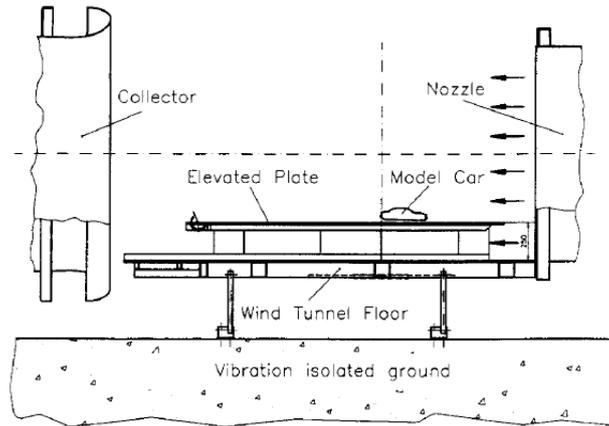


Figure 2.1: Vue de côté de la soufflerie avec le modèle de véhicule.

2.3 Description de Gmsh

Gmsh est un logiciel libre permettant de générer des maillages 2D et 3D de type éléments finis avec des outils de pré/post-traitement. Son objectif de conception est d'offrir un outil de maillage rapide, léger et facile à utiliser avec des données paramétriques et des capacités de visualisation avancées.

Gmsh est construit autour de quatre modules : géométrie, maillage, solveur et post-traitement. Tous ces modules sont prescrits de manière interactive en utilisant l'interface utilisateur graphique ('*GUI*')(Figure 2.2) ou dans des fichiers textes en utilisant son propre langage de programmation. Ceci permet d'automatiser tous les traitements, en utilisant des boucles, conditionnelles et système externe d'appels. Une brève description des quatre modules est donné ci-après.

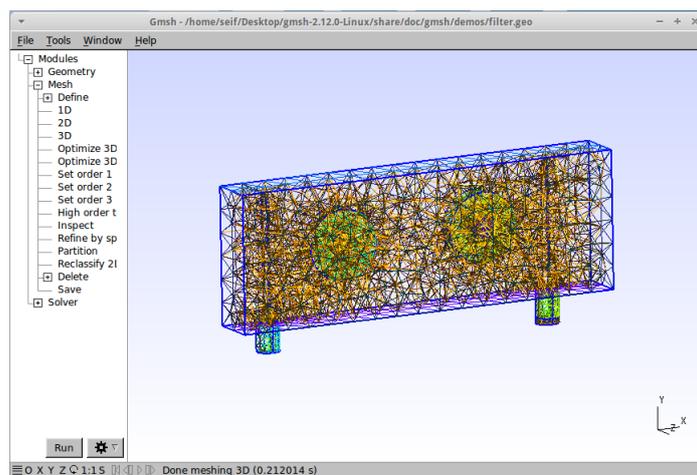


Figure 2.2: Fenêtre principale de Gmsh.

2.3.1 Géométrie dans Gmsh

Gmsh utilise une représentation limite (“BRep”) pour décrire la géométrie. Les modèles sont créés dans un flux ascendant par points définissant successivement, lignes orientées (segments de lignes, cercles, ellipses, splines, . . .), surfaces orientées (surfaces planes, surfaces réglées, surfaces triangulées,...) et des volumes. Des groupes d’entités géométriques (appelés “*groupes physiques*”), peuvent aussi être définis, basés sur ces entités géométriques élémentaires. Son langage de programmation permet que toutes les entités géométriques peuvent être pleinement paramétrées.

2.3.2 Maillage avec Gmsh

La génération de maillage est réalisé dans le même flux ascendant que la création de géométrie : les lignes sont discrétisées en premier ; le maillage des lignes est ensuite utilisé pour mailler les surfaces ; puis le maillage des surfaces est utilisé pour mailler les volumes. Dans ce procédé, le maillage d’une entité est seulement limitée par le maillage de ses limites. Par exemple, en trois dimensions, les triangles discrétisant une surface seront forcés d’être visages de tétraèdres dans le maillage 3D finale seulement et seulement si la surface est située dans la limite d’un volume ; les éléments de ligne discrétisant une courbe seront forcés d’être des bords de tétraèdre dans le maillage 3D finale seulement et seulement si la courbe est située dans la limite d’une surface, elle-même partie de la limite d’un volume ; un nœud unique discrétisation d’un point au milieu d’un volume sera forcé d’être un sommet de l’un des tétraèdre dans le maillage 3D finale seulement et seulement si ce point est connectée à une courbe, elle-même partie de la limite d’une surface, elle-même partie de la limite d’un volume.

Remarque : *Gmsh* est aussi un solveur et un post-traitement, pour plus d’informations visiter son cite officiel.

2.4 Création de la géométrie

Un modèle [4] bidimensionnel de véhicule a été construit (Figure2.3), c’est une forme en coupe transversale d’une voiture de tourisme typique. Les dimensions sont de $L = 298\text{ mm}$ dans la direction longitudinal, $H = 823\text{ mm}$ de hauteur et d’environ $e = 2\text{ m}$ dans le sens de largeur de l’envergure.

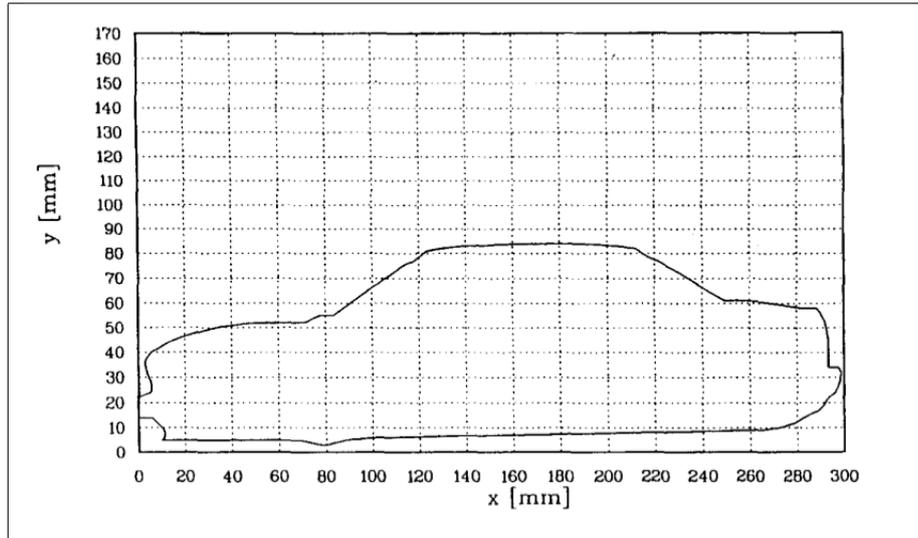


Figure 2.3: Géométrie du modèle de véhicule [? ?].

Le domaine de calcul englobant le modèle est représenté sur la figure 2.4. Les limites d'entrée et de sortie sont situés successivement à une longueur de corps L en amont et $4.66L$ en aval. La limite supérieure est située au-dessus du corps à $2.6L$ et la garde au sol est de $0.1L$.

Nous avons utilisé *Gmsh* pour la création de la géométrie et du domaine de calcul. Les détails sont données en annexe (A). Nous obtenons en finalité le résultat de figure (2.4) ci-dessous :

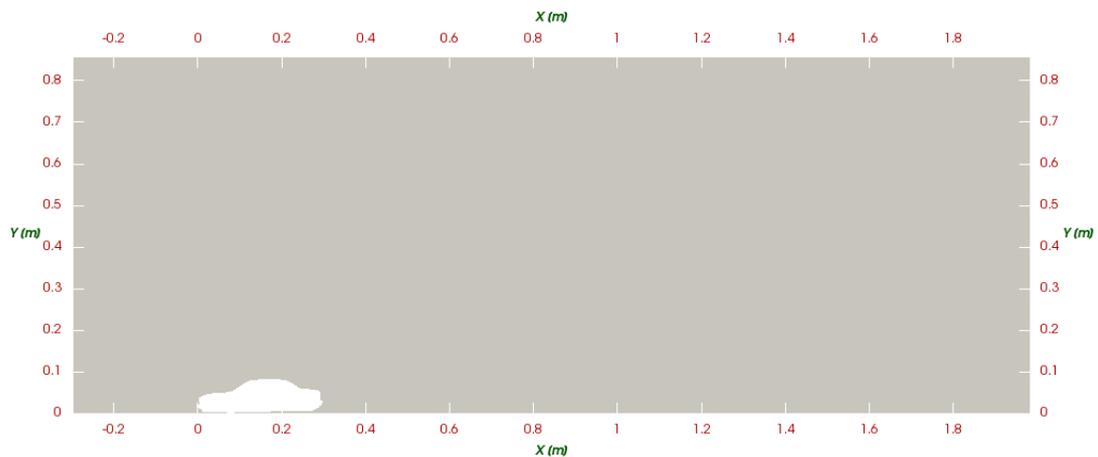


Figure 2.4: Domaine autour du modèle de voiture.

2.5 Maillage du modèle

Le module de maillage de *Gmsh* regroupe plusieurs algorithmes de maillage 1D, 2D et 3D, toutes les grilles produites sont conformes au sens des éléments finis.

- Les algorithmes non structurés 2D génèrent des triangles ou les deux triangles

et quadrilatères (lorsque '*recombiner Surface*' est utilisé). Les algorithmes non structurés 3D génèrent tétraèdres.

- Les algorithmes 2D structurés (de '*transfinis*' et '*extrusion*') génèrent des triangles par défaut, mais des quadrilatères peuvent être obtenus à l'aide des commandes '*Recombine*'. Les algorithmes de structure 3D génèrent des tétraèdres hexaédrique, des prismes, des pyramides, en fonction du type de la surface des mailles sur lequel ils sont basés.

Dans notre étude, nous avons créer 4 types de maillages afin de voir leurs influence sur la solution :

1. Maillage "Box field".
2. Maillage "Ball field".
3. Maillage "Multi-Bloc".
4. Maillage "Hybride".

Les deux premiers étant de types non structurés et les deux dernier de type structurés.

2.5.1 Maillages non structurés

Dans *Gmsh*, on peut spécifier la taille générale de maille "*champ*". Différents champs existent :

Box field , Attractor field, MathEval field, BoundaryLayer field ,.... etc.

2.5.1.1 Maillage avec un champ rectangulaire "Box field"

Les détails permettant d'obtenir ce maillage (Figure 2.5) sont reportés en annexe (B.1).

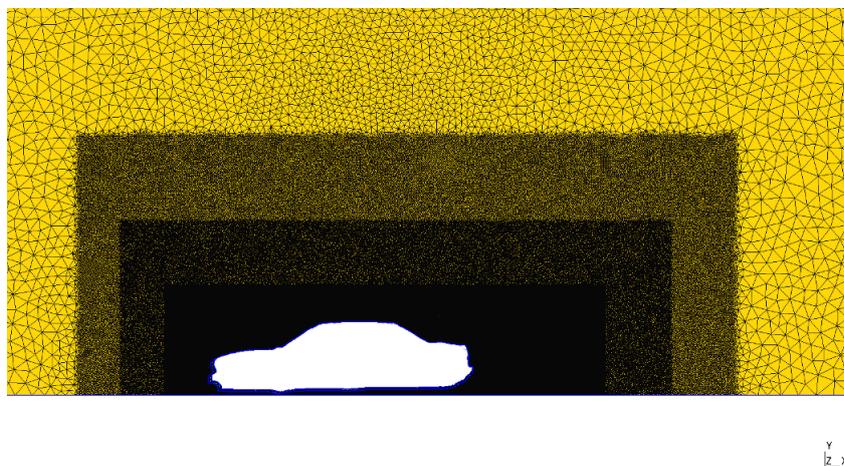


Figure 2.5: *Maillage Box field.*

Caractéristiques du maillage :	Nombre des cellules	178345
	Nombre des noeuds	179677
	Taille sur disque	38 MB

2.5.1.2 Maillage avec un champ circulaire “Ball field”

Les détails permettant d’obtenir ce maillage (Figure 2.6) sont reportés en annexe (B.2).



Figure 2.6: *Maillage Ball field.*

Caractéristiques du maillage :	Nombre des cellules	190016
	Nombre des noeuds	191390
	Taille sur disque	40 MB

2.5.2 Maillage structuré “Multi-Bloc”

Le maillage Multibloc (Figure 2.7) est le plus difficile à réaliser. En effet, la création des différents bloc (annexe B.3) est une lourde tâche et prend énormément de temps. Par contre, l’opération de maillage ne dure pas aussi longtemps que celle des autres types de maillages puisque c’est lui qui possède le plus faible nombre de noeuds.

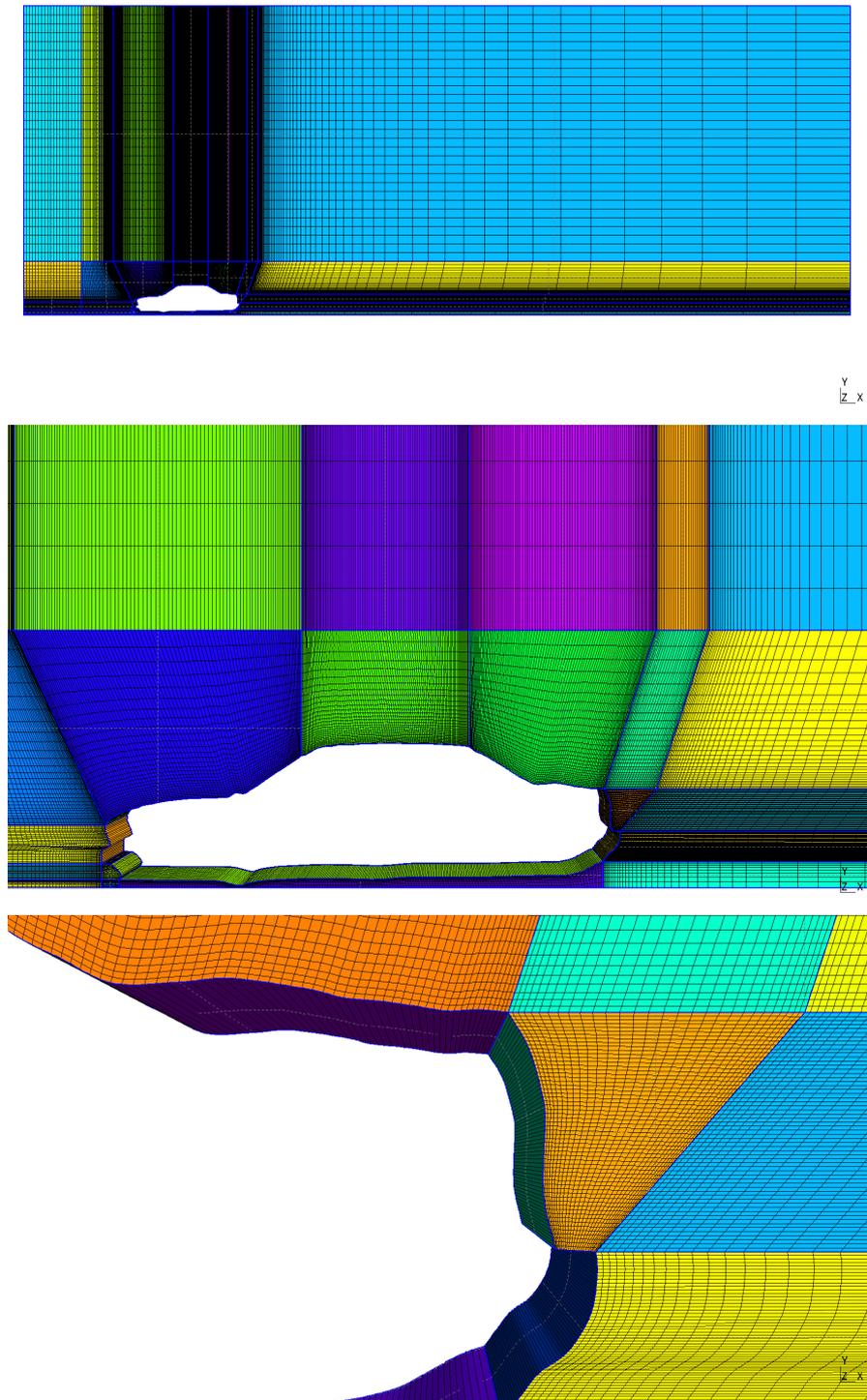


Figure 2.7: *Maillage multi-bloc.*

Caractéristiques du maillage :	Nombre des cellules	35894
	Nombre des noeuds	73804
	Taille sur disque	11 MB

2.5.3 Maillage Hybride

Les détails permettant d'obtenir ce maillage (Figure 2.8) sont reportés en annexe (B.4).

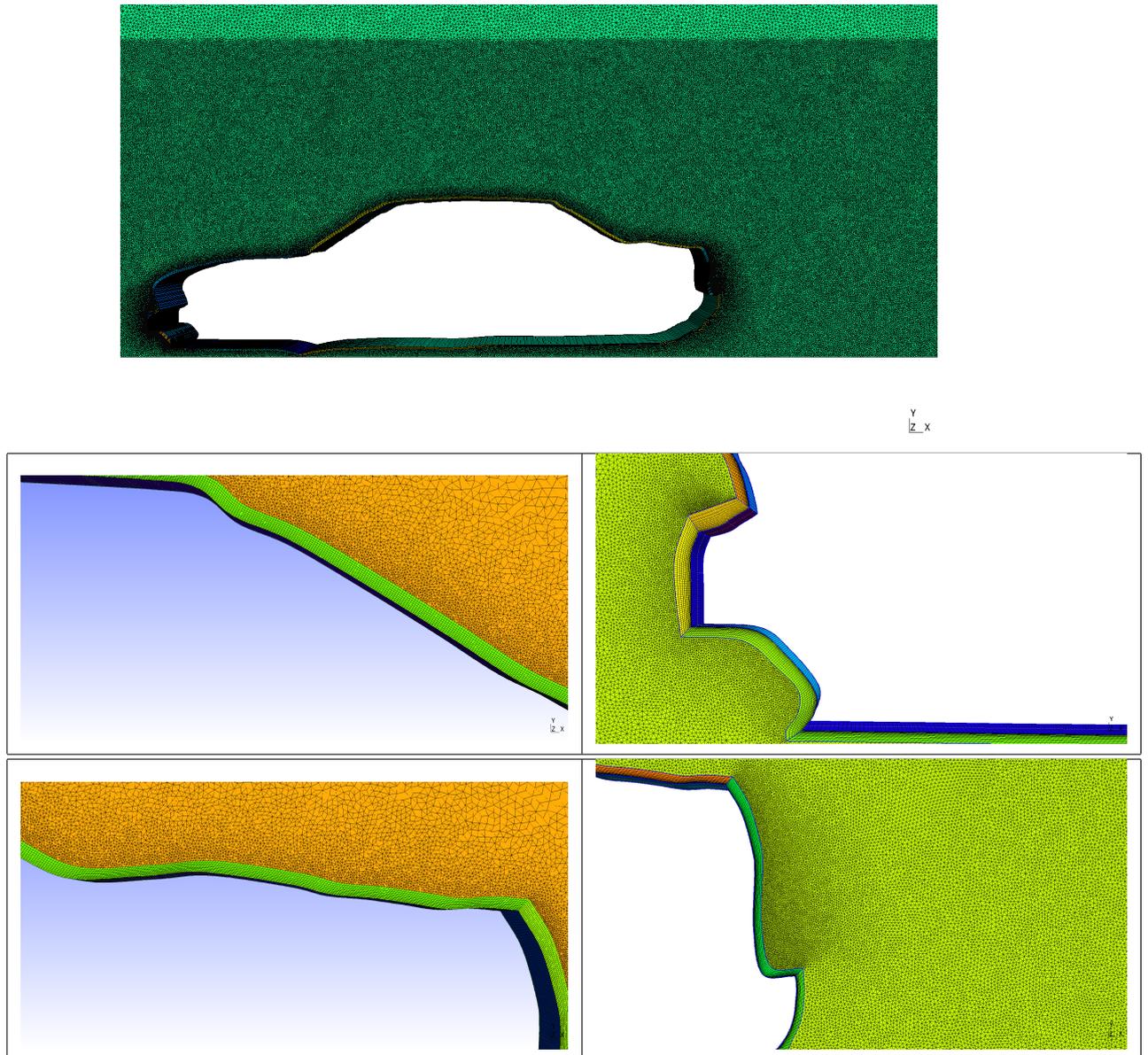


Figure 2.8: *Maillage hybride*

Caractéristiques du maillage :	Nombre des cellules	381099
	Nombre des noeuds	??
	Taille sur disque	85 MB

2.6 Conclusion

Dans ce chapitre, nous avons présenté l'essentiel de notre travail concernant notamment la conception du modèle de véhicule à partir de ces coordonnées prises dans

l'article de référence [4], la conception du domaine de calcul et enfin des différents maillages structurés et non structurés. Tout cela est fait avec le logiciel Opensource *Gmsh*.

Le prochain chapitre sera consacré aux simulations des différents cas en fonction des différents maillages et en variant le modèle de turbulence..

CHAPITRE 3

SIMULATIONS ET RÉSULTATS

3.1 Introduction

Après une brève présentation du code OpenFoam, nous allons, dans ce dernier chapitre, simuler l'écoulement autour du modèle de véhicule déjà créé auparavant. Nous allons en premier lieu comparer les simulations avec les différents maillages réalisés afin d'en extraire le meilleur en utilisant uniquement le modèle de turbulence $k - \epsilon$. Ce choix est dû au fait que nous possédons les résultats expérimentaux pour ce modèle, ce qui nous permettra de comparer nos résultats. Dans un second lieu, nous utiliserons le maillage choisi et nous effectuerons différentes simulations avec d'autres modèles de turbulence, à savoir : $k - \omega$ et SA (*Spalar-Allmaras*).

3.2 Présentation d'OpenFOAM

OpenFOAM est livré avec de nombreux solveurs couvrant une large gamme de domaines tels que la combustion, les écoulements compressibles, incompressibles, multiphasiques, avec réactions chimiques, les transferts thermiques... Différents modèles de turbulence (RANS, LES, ...) sont également présents. *OpenFOAM* est distribué avec *ParaView*, un logiciel de post-traitement open source. Contrairement à la majorité des codes scientifiques écrits de façon séquentielle (souvent en Fortran), *OpenFOAM* profite de la puissance des langages orientés objet (POO). Cette structure sous forme de classes permet de se rapprocher de l'écriture mathématique en termes d'opérateur divergence, rotationnel, gradient, laplacien, dérivée temporelle...etc. Aucune connaissance approfondie du C++ n'est nécessaire pour écrire son modèle dans *OpenFOAM*[12].

3.2.1 Solveurs dans OpenFoam

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 est celle du solveur à utiliser. Contrairement à *Fluent*, *OpenFOAM* a, du point de vue de l'utilisateur, presque autant de solveurs différents que de type de cas d'étude possibles. Par exemple, il a un solveur nommé «*icoFoam*» dédié uniquement aux écoulements incompressibles laminaires instationnaires. Un autre solveur nommé «*simpleFoam*» est dédié aux écoulements incompressibles turbulents stationnaires. Ainsi, il n'y a pas besoin de choisir les équations à résoudre, comme pour *Fluent*, mais uniquement celui de choisir le bon solveur.

Pour nos simulations, nous avons utilisé le solveur «*simpleFoam*».

3.2.2 Dossier de travail

Avec *OpenFOAM*, un dossier de travail (maillage et calcul) est celui sur lequel l'utilisateur est en train de travailler. Sa structure est donnée sous forme d'arbre dans la figure suivante.



Figure 3.1: Structure du dossier de travail sous *OpenFoam*.

Cette structure n'est pas figée et est susceptible d'être étoffée pendant l'utilisation. Néanmoins, la base est toujours la même.

Dossiers temporels : 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 turbulente, ... etc) réunies dans des fichiers dont le nom correspond à ces grandeurs (exp. «**p**» pour la pression, «**U**» pour la vitesse, «**k**» pour l'énergie cinétique turbulente). Le nom de ces dossiers temporels correspond au temps en secondes associé à ces valeurs (exp. «0», «1», «0.1253»). 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. On note que le format du nom des dossiers temporels peut être modifié dans le fichier *system/controlDict* [11].

Dossier “ system ” : Le dossier «*system*» est un dossier fondamental qui contient un fichier utilisateur «*controlDict*» dans lequel celui-ci spécifie les contraintes temporelles du calcul (temps initial, temps final, intervalle de temps entre deux calculs, intervalle entre deux dossiers temporels, etc.). Le dossier «*system*» contient également deux fichiers utilisateurs «*fvSchemes*» et «*fvSolution*» dans lesquels l'utilisateur paramètre les algorithmes de résolution utilisés. Ces trois fichiers sont indispensables à la lecture du cas sous *ParaView* ainsi que pour exécuter les commandes *OpenFOAM* et doivent être les premiers fichiers présents lors de la création d'un nouveau cas d'étude. Les fichiers «*fvSchemes*» et «*fvSolution*» sont habituellement copiés depuis le dossier «*tutorials*» d'*OpenFOAM*. Bien qu'ils soient peu intuitifs pour le débutant, leur importance est capitale et il est indispensable de savoir s'en servir pour utiliser *OpenFOAM*. Ce dossier contient aussi, le cas échéant, les fichiers utilisateurs «*snappyHexMeshDict*» et «*decomposeParDict*» qui permettent d'utiliser l'algorithme de modification de maillage avec la commande «*snappyHexMesh*». Comme pour «*fvSchemes*» et «*fvSolution*», le fichier «*decomposeParDict*» peut être copié depuis un dossier de cas du dossier «*tutorials*».

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 du même type. Ces fichiers ne sont pas indispensables à la lecture du cas sous *ParaView* et leur présence dépend 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*» qui permet de créer un maillage structuré de type «*blockMesh*». Le dossier «*constant*» peut également contenir un dossier «*triSurface*» qui contient la ou les géométrie(s) surfacique(s) STL dans le cas d'une modification de maillage via la commande «*snappyHexMesh*» [11].

3.2.3 Conversion lu maillage

Plusieurs packages sont disponibles sous OpenFoam et qui permettent les conversion des différents maillages. Les commandes sont :

- *fluentMeshToFoam* permettant la conversion depuis *Fluent* (2-D et 3-D) écrit en format “.msh”.
- *starToFoam* permettant la conversion depuis *STAR-CD* / *PROSTAR*.
- *gambitToFoam* permettant la conversion depuis *GAMBIT* “.neu”.
- *ideasToFoam* permettant la conversion depuis *IDEAS* écrit en format *ANSYS* “.ans”.

- *cfx4ToFoam* permettant la conversion depuis *CFX* écrit en format *.geo*.
- *gmshToFoam* lit les fichier de maillage de *Gmsh* en format *.msh*.

Nous utiliserons donc cette dernière commande afin de convertir tous nos fichier créés avec *Gmsh*.

3.3 Influence du types de maillage

Nous allons comparer les différents maillages (*Boxfield*, *Ballfield*, *Multibloc* et *Hybrid*) et nous allons en déduire le meilleur en utilisant le modèle de turbulence $k - \epsilon$.

3.3.1 Simulations

Afin de ne pas allourdir le document pour la lecture, nous renvoyons le lecteur, chaque fois que c'est nécessaire, vers les différentes annexes où nous avons mis les détails nécessaire.

Pour cette simulation, l'annexe (C) nous indique les différentes étapes.

Afin de comparer nos résultats aux résultats expérimentaux de la référence [4], nous avons tracé les courbes aux mêmes positions qui sont indiquées (Figure 3.2).

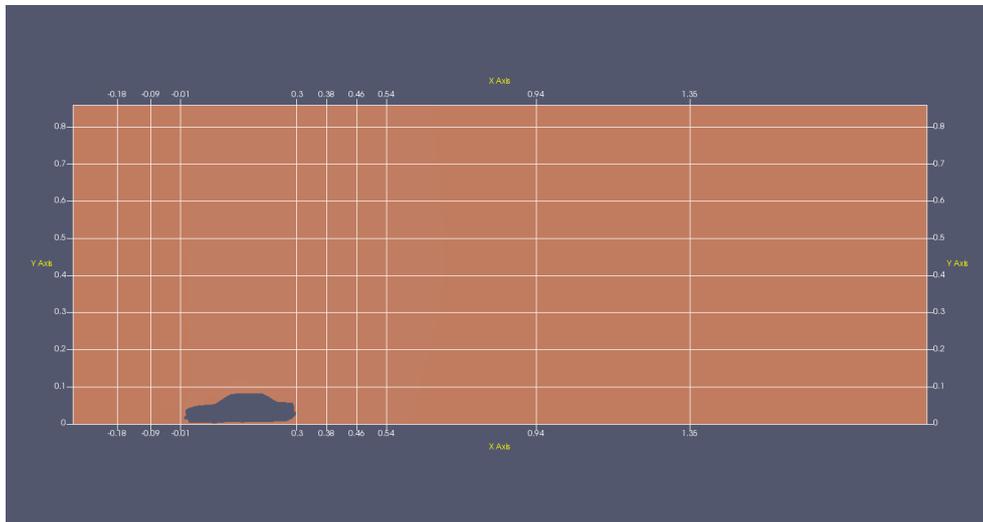
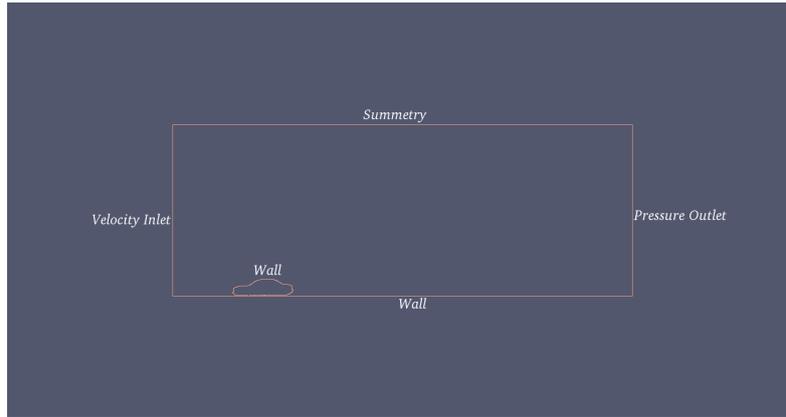


Figure 3.2: Positions des stations de mesures expérimentales ($\frac{x}{L}$).

Les types de conditions aux limites prescrites sur les interfaces du domaine de calcul sont indiqués sur la figure (3.3) :

Figure 3.3: *Types de conditions aux limites.*

3.3.2 Résultats

Pour visualiser les résultats nous avons utilisé les outils *OpenSource* suivants :

ParaView : C'est un logiciel libre de visualisation scientifique de données (2D et 3D). Il est fondé sur la bibliothèque VTK et publié sous licence BSD. Il est développé principalement par le *Sandia National Laboratories*, le *Los Alamos National Laboratory* et la société *Kitware Inc.* *ParaView* dispose d'une architecture client-serveur qui permet de traiter des données à distance ; typiquement, pour un volume de données important, tout ou partie du traitement est effectué par un serveur, le poste de travail servant uniquement à l'affichage. En visualisation volumique, il réduit le niveau de détail des objets éloignés afin de maintenir une bonne vitesse d'affichage [16]. C'est outils très performant qui a été conçu principalement pour traiter de grandes quantités de données et pour un travail parallèle.

QtiPlot : C'est aussi une plate-forme de visualisation scientifique 2D et 3D. Il est idéal pour les étudiants et propose une alternative aux logiciels propriétaires extrêmement coûteux comme *Origine*, *SigmaPlot*, *SPSS*, *Regressi* ou *Igor Pro*. *QtiPlot* est utilisé pour l'enseignement ainsi que pour l'analyse de données complexes et la visualisation dans les entreprises, les lycées, les universités et les instituts de recherche du monde entier. On trouvera dans la référence [14] une courte liste d'ouvrages scientifiques citant *QtiPlot*. Notons que nous avons utilisé ce logiciel uniquement pour les résultats 2D.

- Nous allons exploiter les résultats que nous avons trouvé durant les simulations que nous avons effectué sous *OpenFOAM*.

Nous présentons (Figure 3.4) l'évolution de la composante axiale de la vitesse en fonction du nombre d'itération en un point M du domaine de calcul.

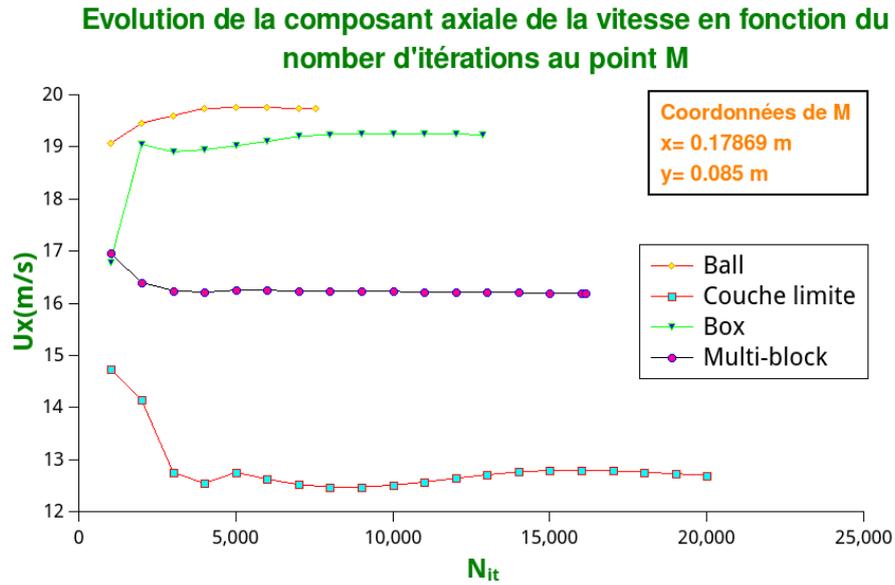


Figure 3.4: Evolution de la composant axiale de la vitesse en fonction du nombre d'itérations.

Ces courbes sont utilisées uniquement pour stopper les itérations quand la valeur de la vitesse est stable. En effet, on ne peut pas dire que le maillage 'Ball field' est meilleur puisque son nombre d'itérations est faible. La figure (3.5) nous montre, au contraire, que le maillage 'Multibloc' est le meilleur puisqu'il converge plus rapidement que les autres.

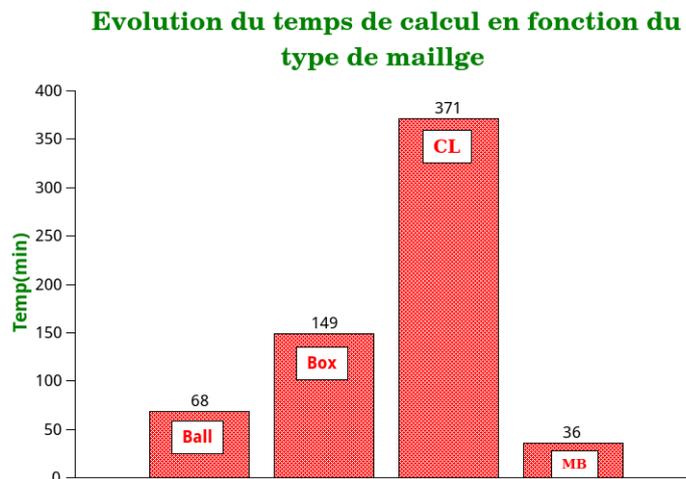


Figure 3.5: Evolution du temps de calcul.

Les coefficients de traînée que nous avons obtenu pour chaque type de maillage sont :

Type de maillage	Box	Ball	MB	CL (Hybride)
Coefficient de traînée	0.262	0.249	0.408	0.478

Tableau 3.1: Coefficient de traînée.

Nous n'avons malheureusement aucune référence qui nous permettra de juger ces résultats.

3.3.2.1 Répartition de vitesse

Les répartitions de vitesse autour du véhicule pour chaque maillage sont représentées ci-dessous :

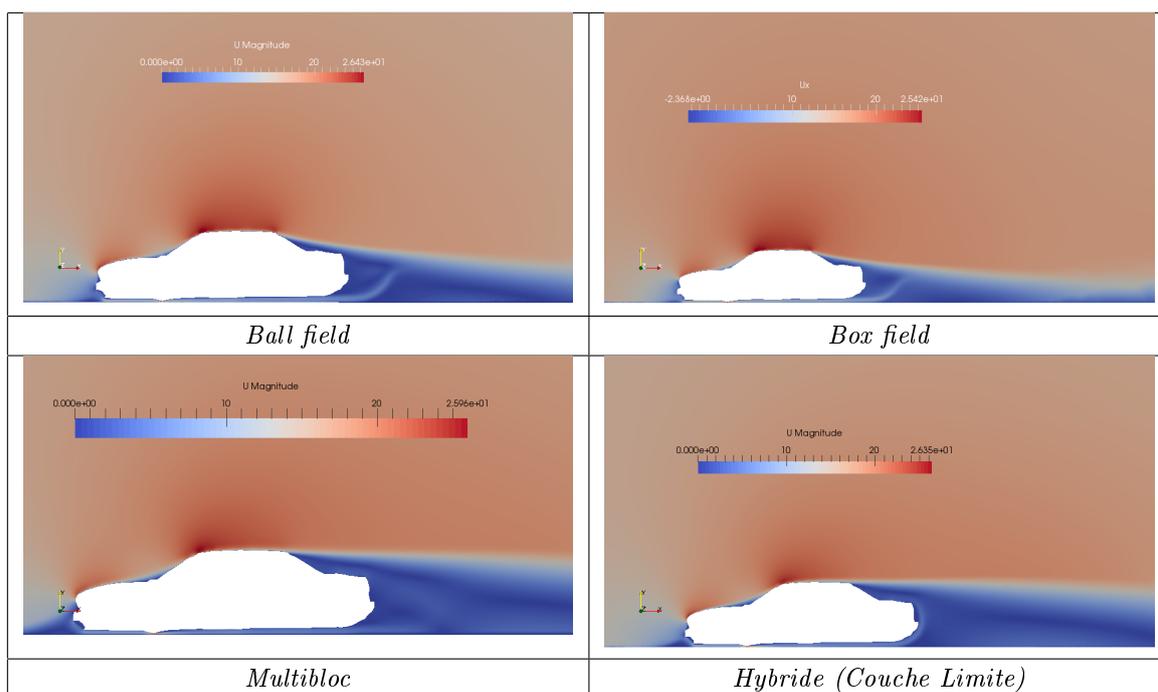


Figure 3.6: Répartition de la vitesse autour du véhicule.

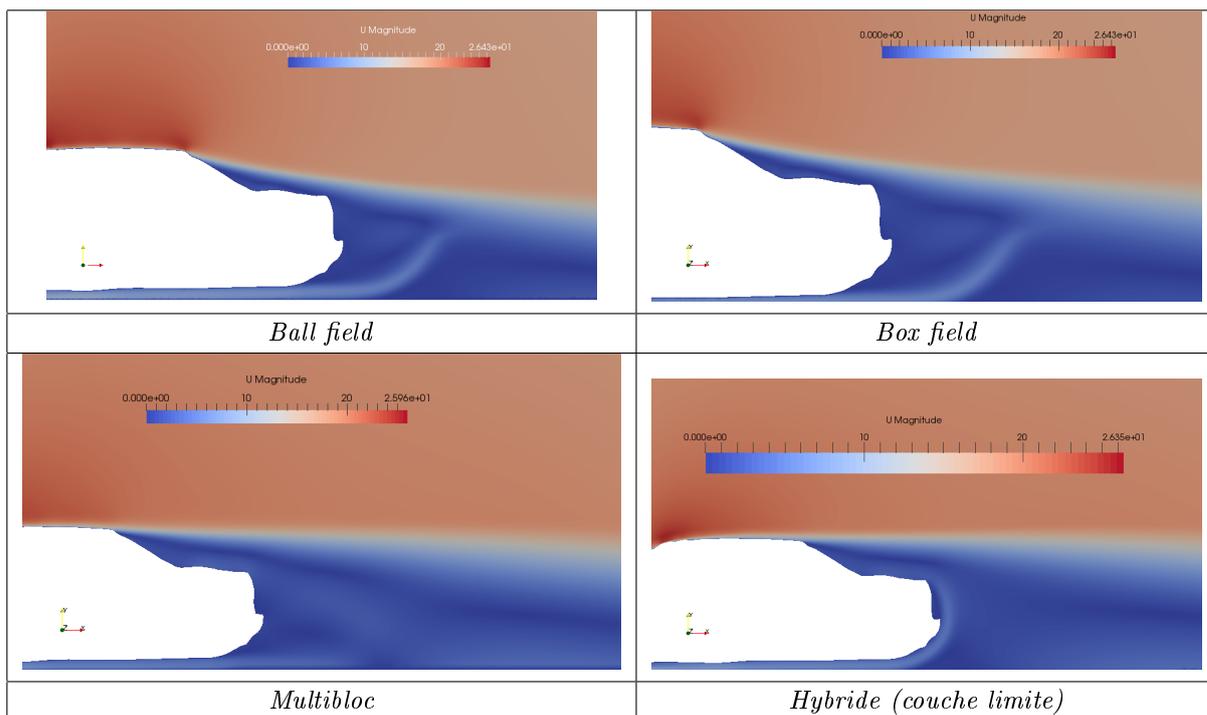


Figure 3.7: Répartition de vitesse derrière le véhicule.

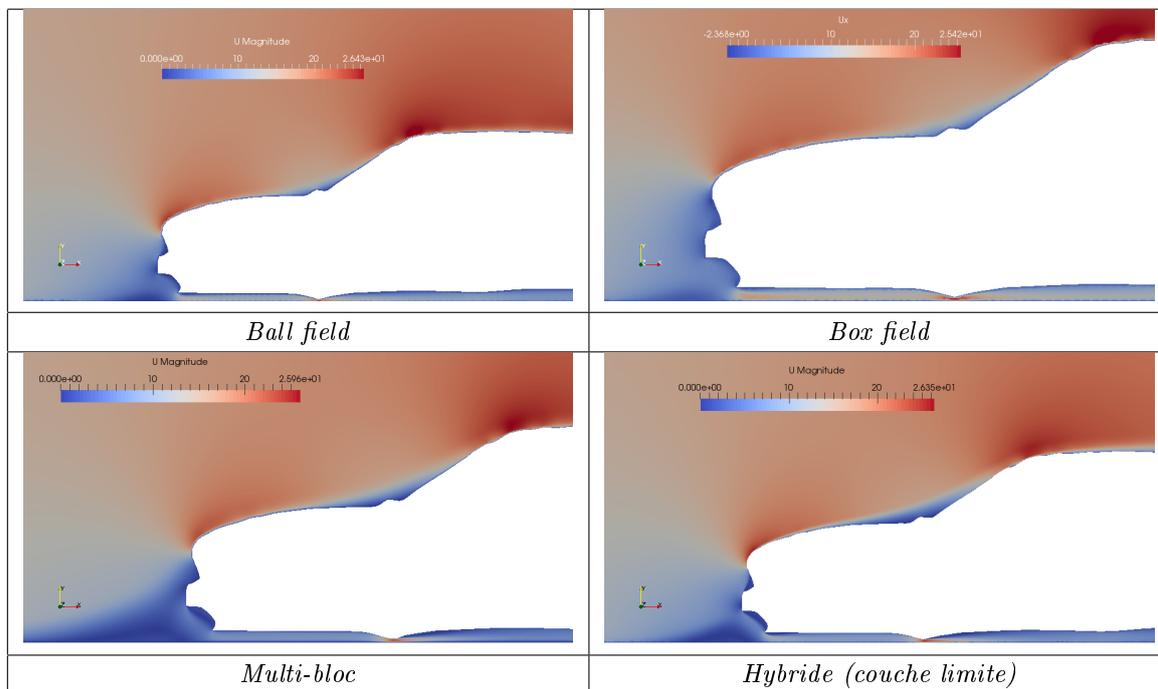


Figure 3.8: Répartition de vitesse devant le véhicule.

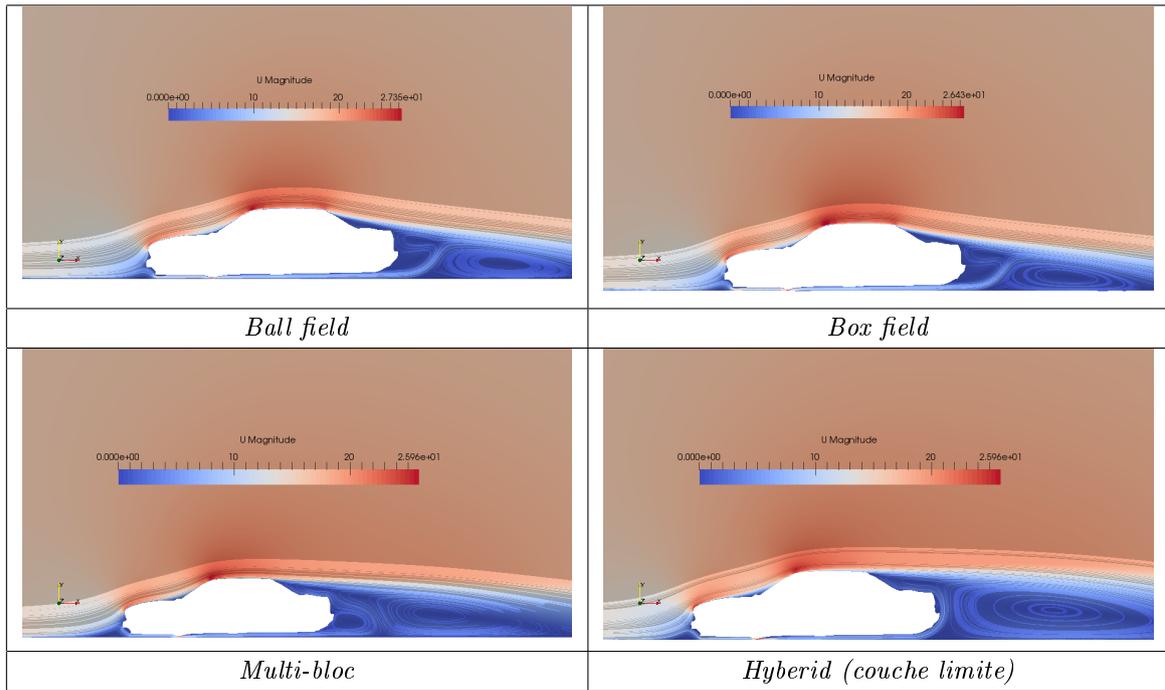


Figure 3.9: Lignes de courant autour du véhicule.

3.3.2.2 Répartition de pression

Les répartitions de pression autour du véhicule pour chaque maillage sont représentées ci-dessous :

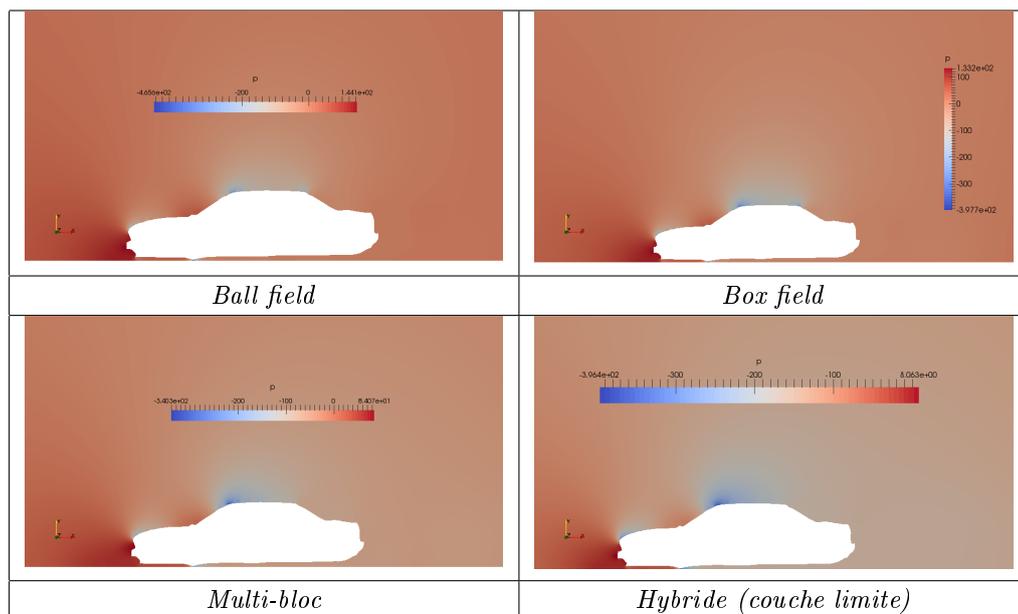


Figure 3.10: Répartition de pression autour du véhicule.

3.3.2.3 Variation de k

Les variations de l'énergie cinétique turbulente k derrière le véhicule pour chaque maillage sont représentées ci-dessous :

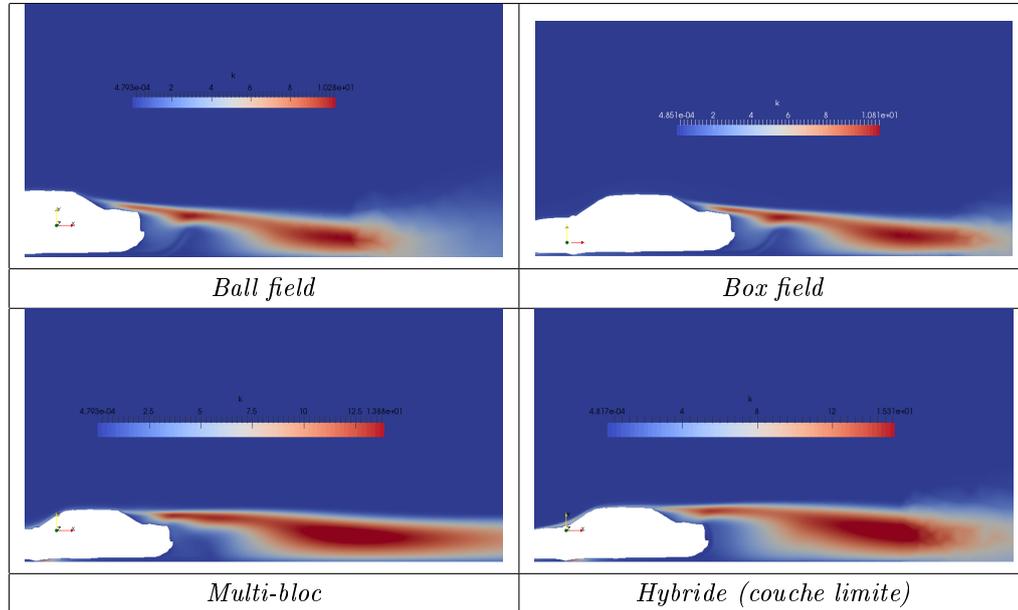


Figure 3.11: Variation de l'énergie cinétique turbulente derrière le véhicule.

3.3.2.4 Comparaison

Les figures (3.12) et (3.13) représentent la comparaison entre les résultats numériques de chaque type de maillage avec les mesures expérimentales pour la composante horizontale de la vitesse U_x dans les différentes stations $\frac{x}{L}$.

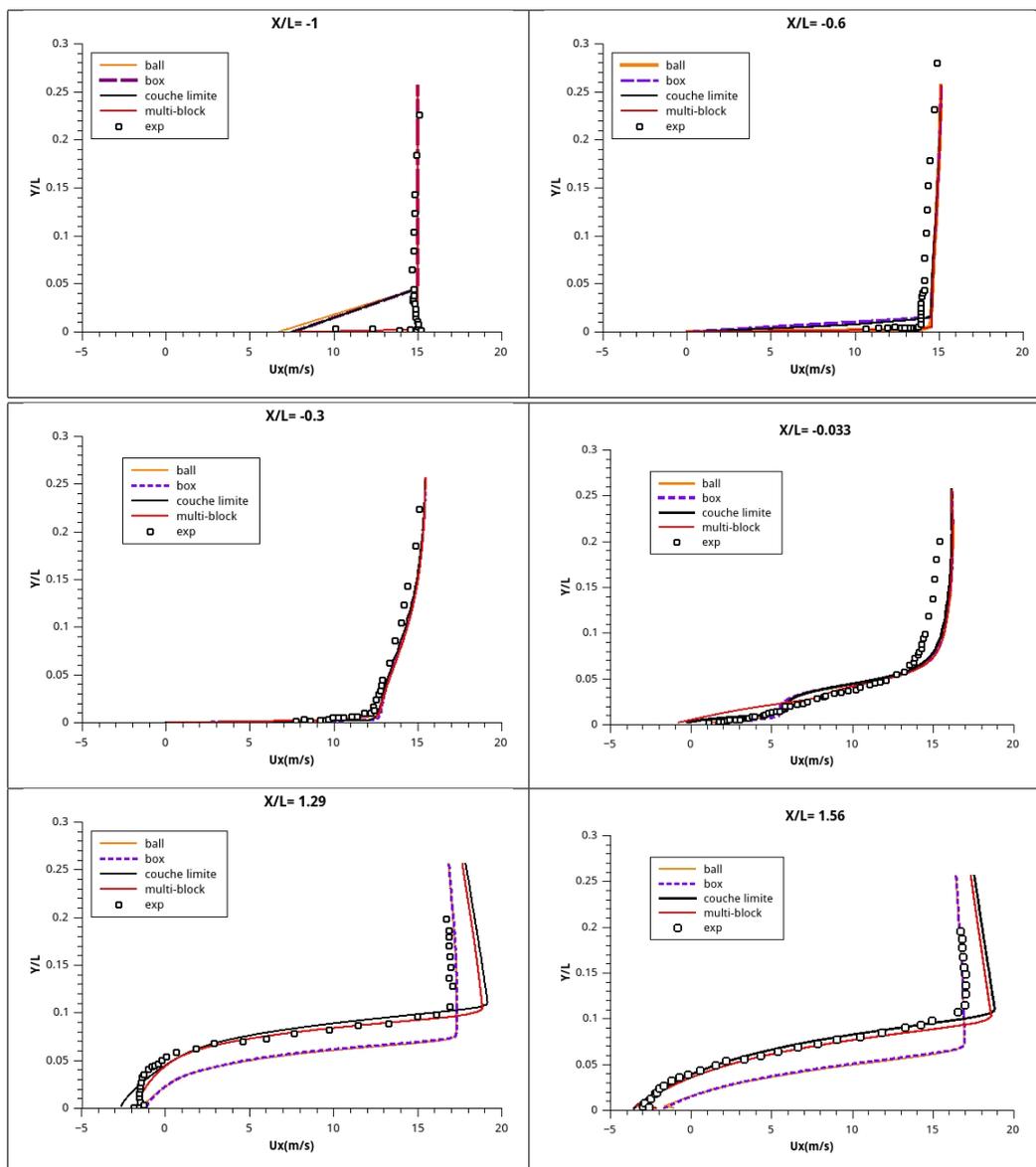


Figure 3.12: Comparaison des vitesses U_x des différents maillages dans différentes stations.

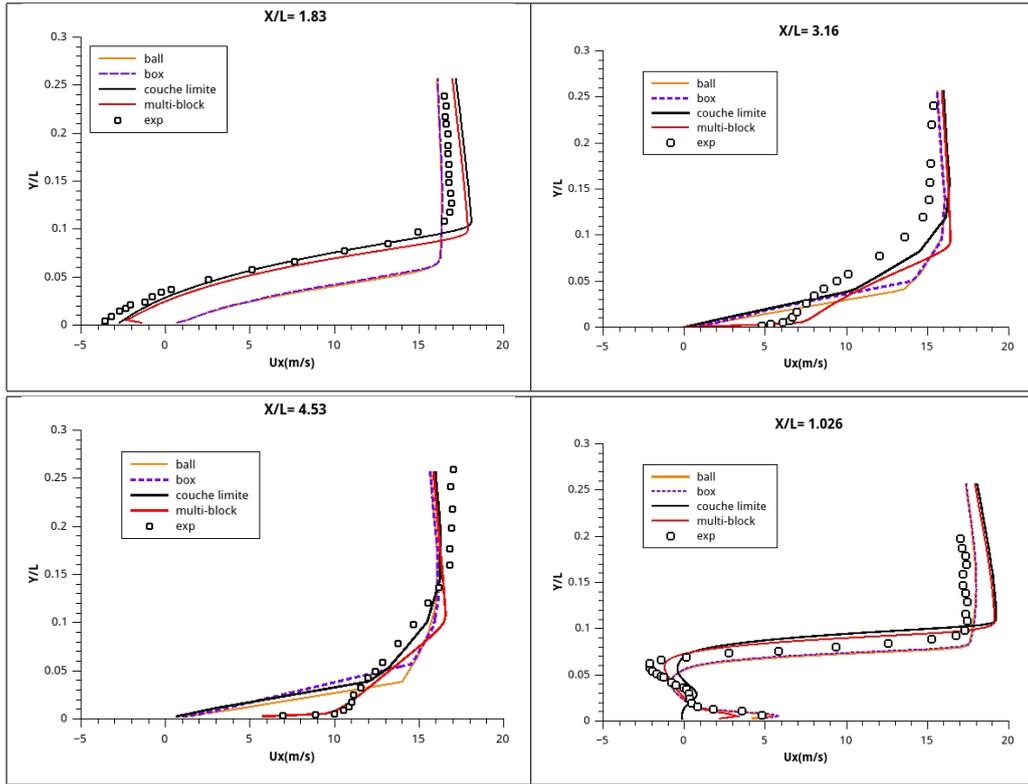


Figure 3.13: Comparaison des vitesses U_x des différents maillages dans différentes stations (suite).

Les figures (3.14) et (3.15) représentent la comparaison entre les résultats numériques de chaque type de maillage avec les mesures expérimentales pour la composante verticale de la vitesse U_y dans les différentes stations $\frac{x}{L}$.

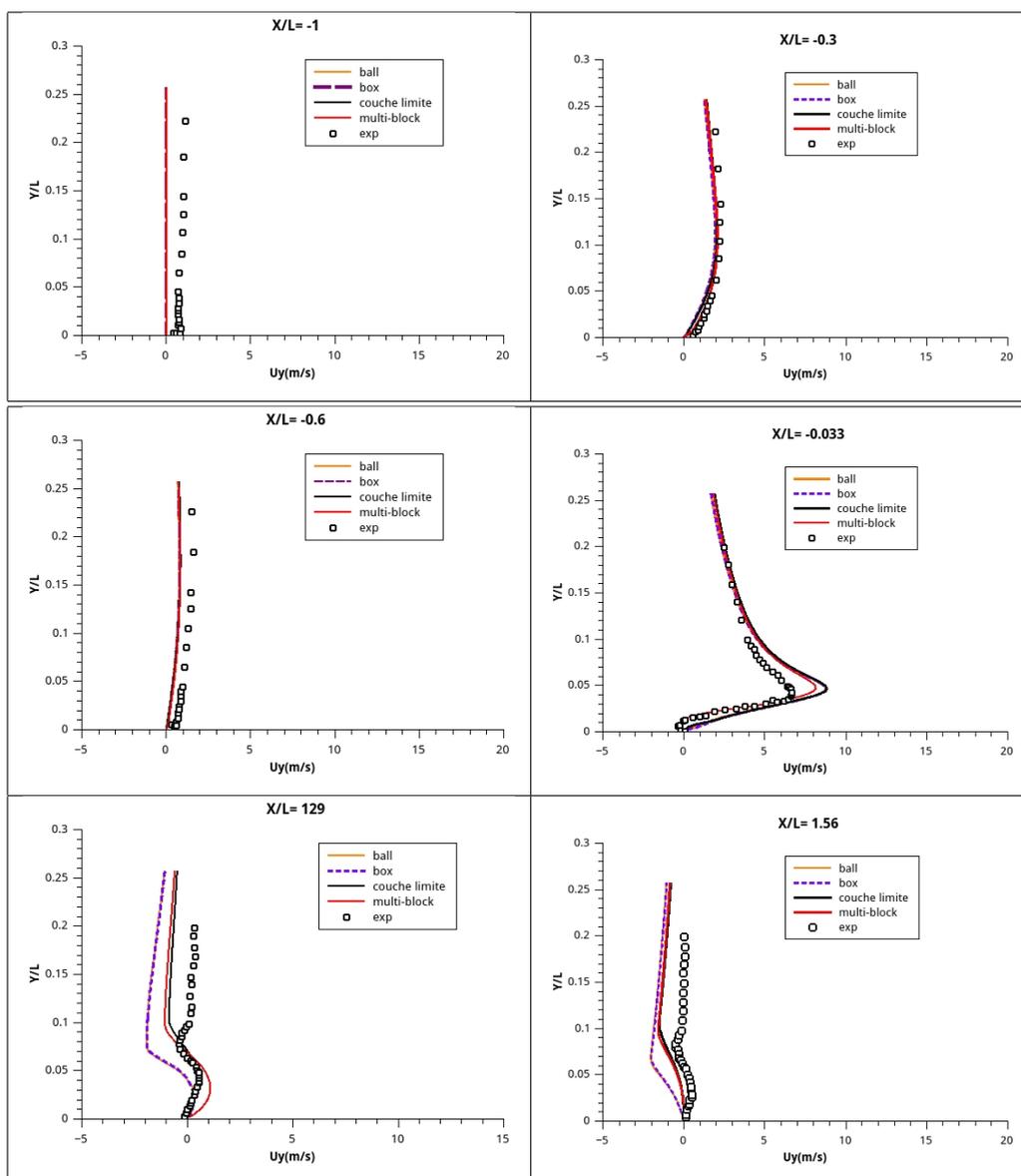


Figure 3.14: Comparaison des vitesses U_y des différents maillages dans différentes stations.

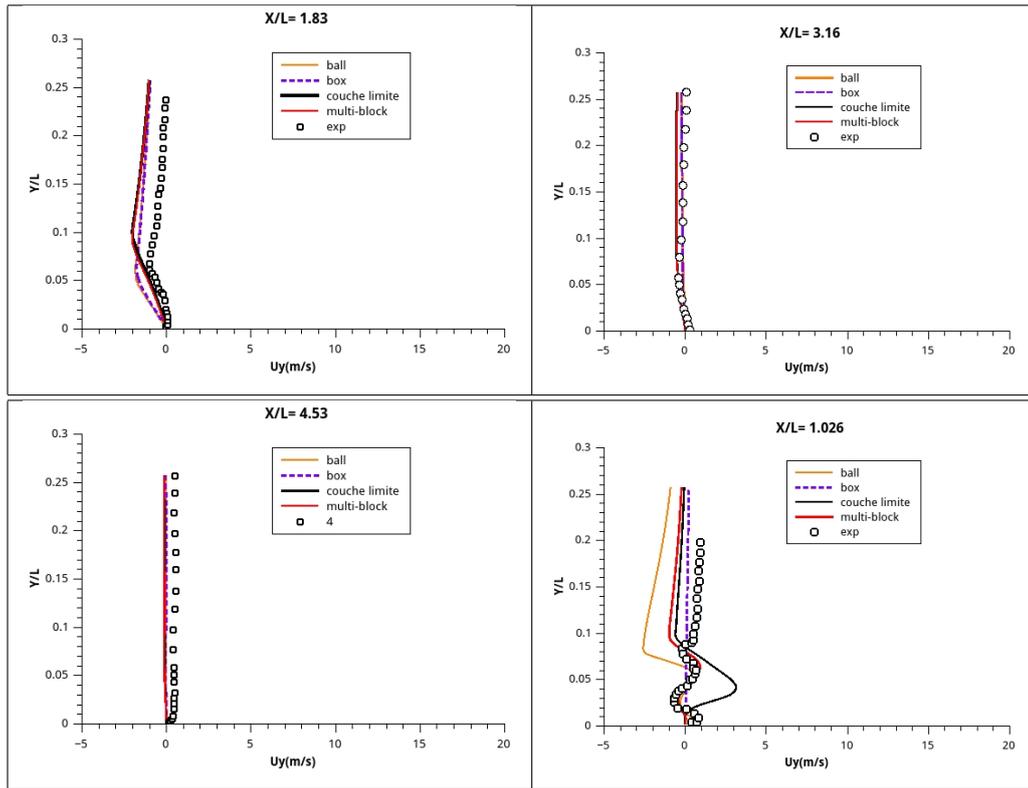


Figure 3.15: Comparaison des vitesses U_y des différents maillages dans différentes stations (suite).

Après examen de tous ces résultats, nous pouvons dire que, bien que les deux maillages 'Multibloc' et 'Hybride' donnent des résultats presque similaires et proches de ceux de l'expérience, le premier l'emporte sur le second pour sa rapidité de convergence (36 mn contre 6 heures !).

3.4 Influence des modèles de turbulence

Dans cette partie, nous allons utiliser le maillage 'Multibloc' que nous avons choisi pour comparer les modèles de turbulence $k - \varepsilon$ standard, $k - \omega$ standard et *Spalart - Allmaras (SA)*.

3.4.1 Simulations

Les étapes de la simulation avec OpenFoam ont déjà été décrites et nous passons directement à l'exploitation des résultats.

3.4.2 Résultats

3.4.2.1 Répartition de vitesse

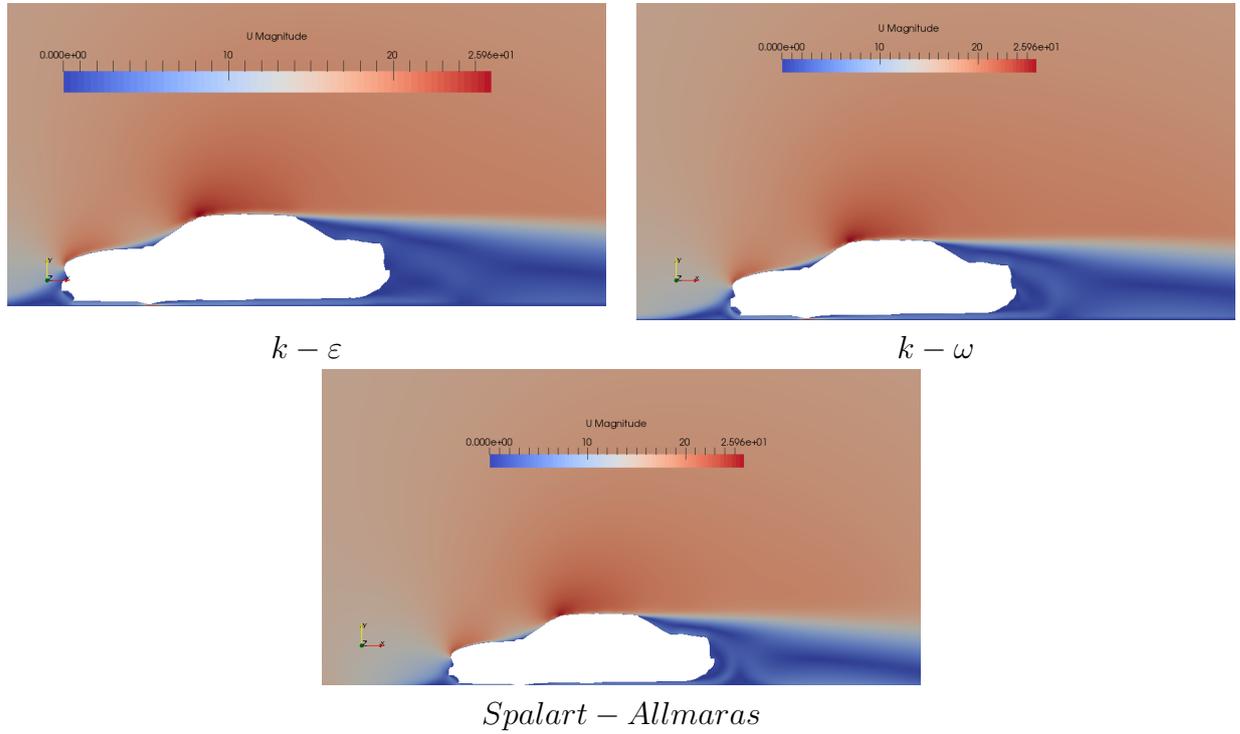


Figure 3.16: Répartition de vitesse autour du véhicule.

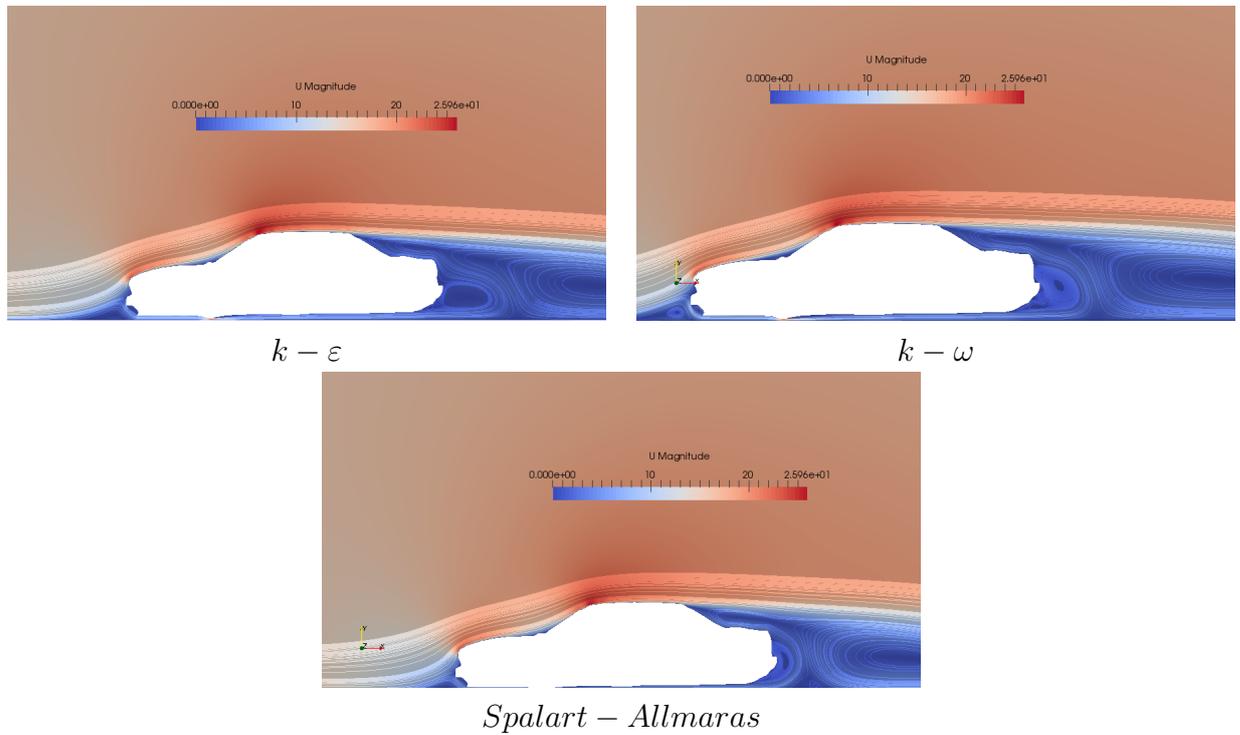


Figure 3.17: Lignes de courant autour du véhicule.

3.4.2.2 Répartition de pression

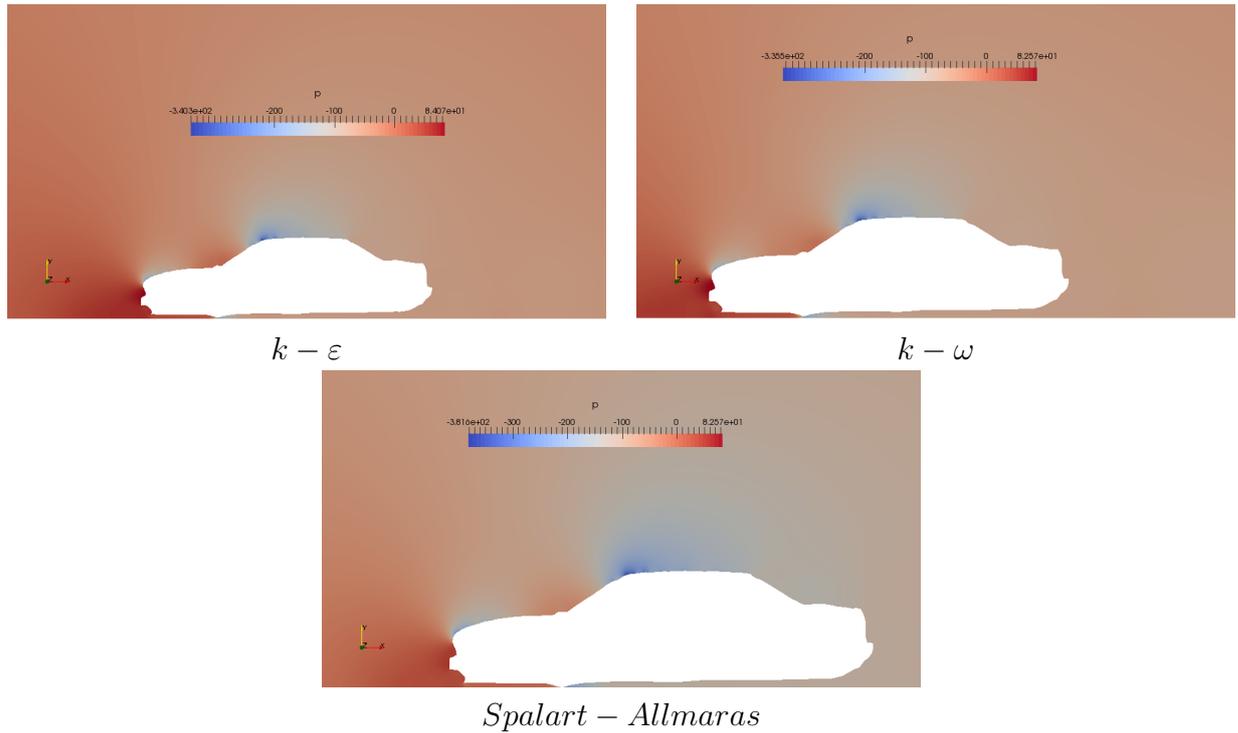


Figure 3.18: Répartition de pression autour du véhicule.

3.4.2.3 Variation de k

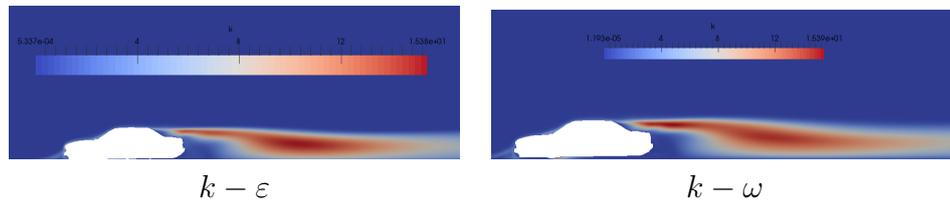


Figure 3.19: Variation de l'énergie cinétique turbulente derrière la véhicule

3.4.3 Comparaison

Les figures (3.20) et (3.21) représentent la comparaison entre les résultats numériques de chaque modèle de turbulence avec les mesures expérimentales pour la composante horizontale de la vitesse U_x dans les différentes stations $\frac{x}{L}$.

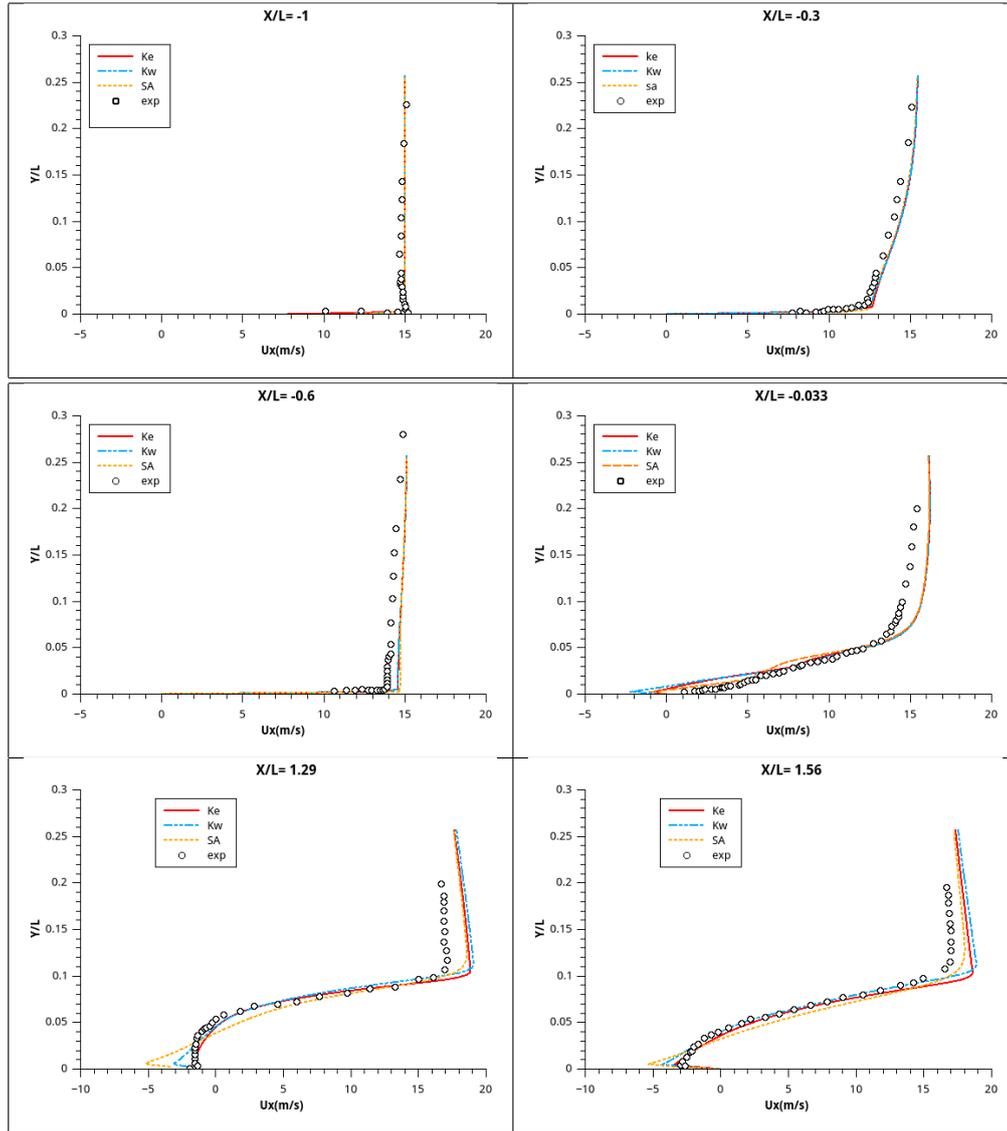


Figure 3.20: Comparaison des vitesses U_x des différents modèles de turbulence dans différentes stations.

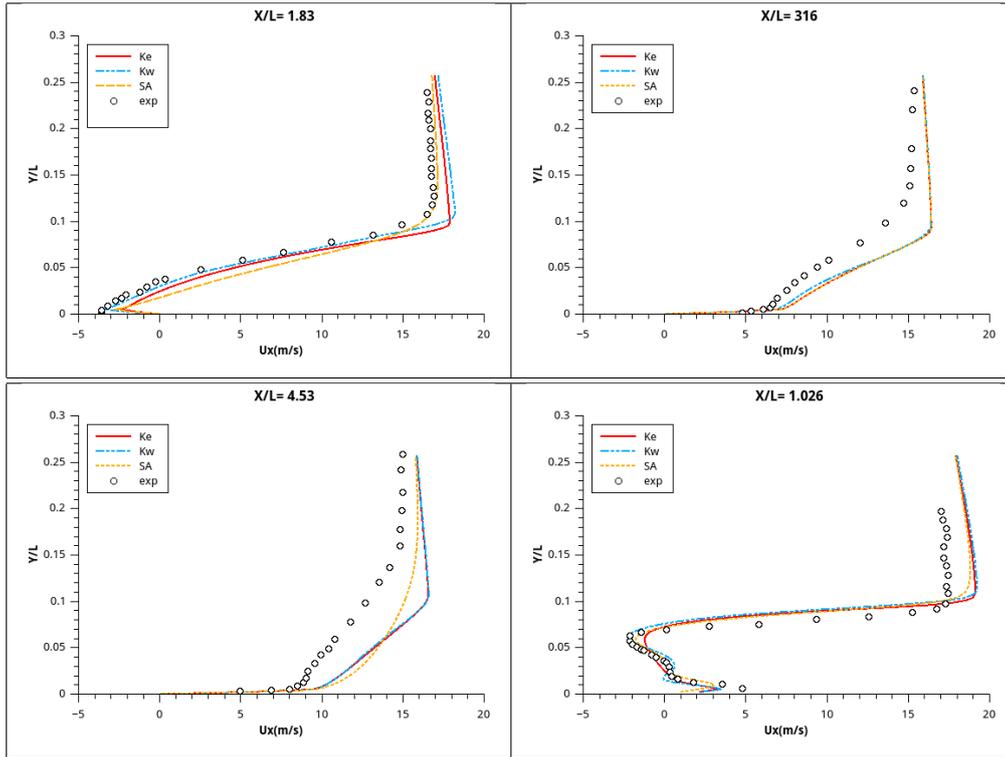


Figure 3.21: Comparaison des vitesses U_x des différents modèles de turbulence dans différentes stations (suite).

Les figures (3.22) et (3.23) représentent la comparaison entre les résultats numériques de chaque modèle de turbulence avec les mesures expérimentales pour la composante verticale de la vitesse U_x dans les différentes stations $\frac{x}{L}$.

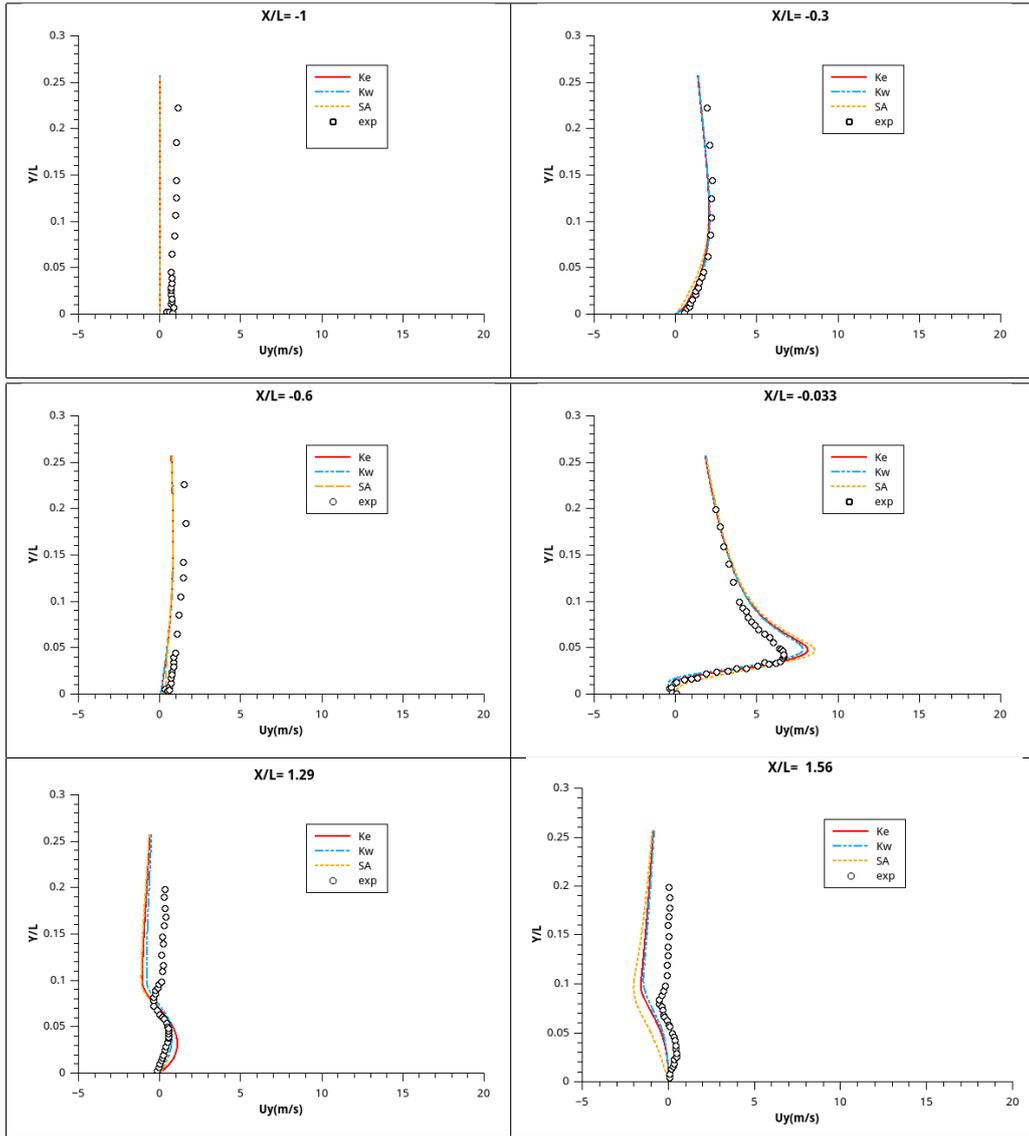


Figure 3.22: Comparaison des vitesses U_y des différents modèles de turbulence dans différentes stations.

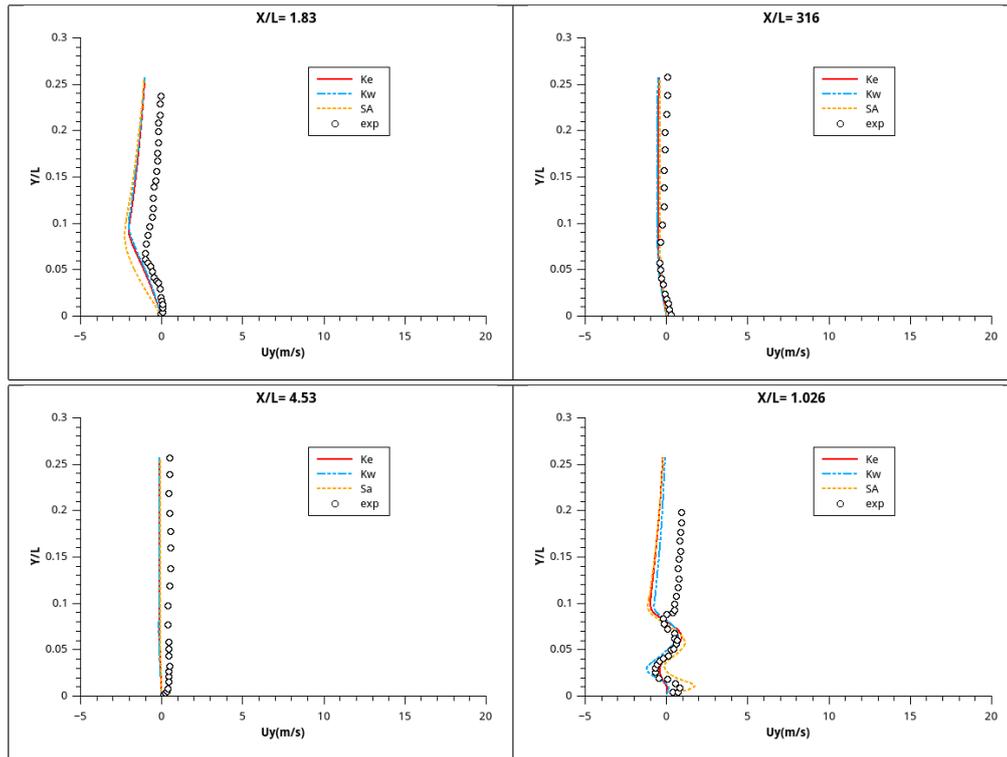


Figure 3.23: Comparaison des vitesses U_y des différents modèles de turbulence dans différentes stations.

Après examination de tous ces résultats, Il nous est très difficile de faire ressortir le modèle de turbulence le plus performant pour ce cas d'étude. En effet, en fonction des différentes zones (avant, juste derrière ou loin derrière le véhicule) les trois modèles donnent des valeurs plus ou moins proches de ceux de l'expérience.

3.5 Conclusion

Dans ce chapitre, nous avons comparé les résultats des simulations faites avec les différents maillages et avec le modèle de turbulence $k - \epsilon$. Nous avons pu ressortir du lot le maillage 'Multibloc' comme étant le plus optimisé tant au niveau précision qu'au niveau temps de calcul. Nous avons ensuite utilisé ce maillage pour simuler l'écoulement autour du modèle de véhicule avec les modèles de turbulence $k - \omega$ et SA (Spalar-Allmaras). Cette comparaison avec le modèle $k - \epsilon$ ne nous a pas permis de choisir lequel des ces trois est le plus performant vu que les résultats obtenus sont très proches.

CONCLUSION GÉNÉRALE

La simulation numérique bidimensionnelle qui a été présentée dans ce travail et qui est autour d'un modèle de véhicule basé sur des résultats expérimentaux de l'article [4] avec l'utilisation des différents types de maillage et de modèles de turbulences ($k-\epsilon$, $k-w$, Spalart-Allmaras). Les conclusions abouties de cette simulation sont présentées comme suit :

- Le maillage qui a une influence très importante dans ce cas est le maillage structuré multi-bloc. Malgré la difficulté rencontrée lors l'exécution de sa génération, il est économique en nombre d'éléments (présente un nombre inférieur de mailles par rapport à un maillage non structuré équivalent). Outre, sa rapidité en terme du temps de calcul, donnant des résultats plus précis et réduit la probabilité d'erreurs numériques.
- La comparaison des modèles de turbulence $k - \omega$ et SA (Spalar-Allmaras) avec le modèle $k - \epsilon$ ne nous a pas permis de choisir lequel des ces trois est le plus performant vu que les résultats obtenus sont très proches.

Finalemant, ce travail offrant la possibilité d'utiliser et exploiter des logiciels opensource benefiques sous le système linux (Gmsh , openFoam, paraview, Qtiplot, Latex, Lyx , Libre Office,)

BIBLIOGRAPHIE

- [1] SR Ahmed. Wake structure of typical automobile shapes. *Journal of Fluids Engineering*, 103(1) :162–169, 1981.
- [2] SR Ahmed. Influence of base slant on the wake structure and drag of road vehicles. *Journal of fluids engineering*, 105(4) :429–434, 1983.
- [3] SR Ahmed, G Ramm, and G Faitin. Some salient features of the time-averaged ground vehicle wake. Technical report, Society of Automotive Engineers, Inc., Warrendale, PA, 1984.
- [4] W Angelis, D Drikakis, F Durst, and W Khier. Numerical and experimental study of the flow over a two-dimensional car model. *Journal of wind engineering and industrial aerodynamics*, 62(1) :57–79, 1996.
- [5] ANSYS. <http://www.ansys.com/products/fluids>.
- [6] PW Bearman. Some observations on road vehicle wakes. Technical report, Society of Automotive Engineers, Inc., Warrendale, PA, 1984.
- [7] PW Bearman, JP Davis, and JK Harvey. Wind tunnel investigation of vehicle wakes. In *Proc. Vehicle Aerodynamics Symp., sponsored by Volkswagenwerk AG-Research Division*, 1982.
- [8] K Hanjalic. One-point closure models for buoyancy-driven turbulent flows. *Annual Review of Fluid Mechanics*, 34(1) :321–347, 2002.
- [9] Wolf-Heinrich Hucho. *Aerodynamics of road vehicles : from fluid mechanics to vehicle engineering*. Elsevier, 2013.
- [10] VERY Chloè LANTENET Johanna. StarCCM+ :étude de l’allée de von karman, travaux des élèves-ingénieurs de la deuxième année de l’enseeiht, 2013.
- [11] H Nouri and F Ravelet. Introduction à la simulation numérique des écoulements. application au transfert thermique sur plaque plane avec starccm+. *DynFluid-Arts et Metiers ParisTech*, 2012.

- [12] CFD Open. Openfoam user guide. *OpenFOAM Foundation*, 2(1), 2011.
- [13] Bruce Perens. La définition de l'open source. *DiBona et alii*, 1999.
- [14] QtiPlot. Data Analysis and Scientific Visualisation, <http://www.qtiplot.com/>.
- [15] M Rauser and J Eberius. Verbesserung der fahrzeugaerodynamik durch unterbodengestaltung. *ATZ. Automobiltechnische Zeitschrift*, 89(10) :535–542, 1987.
- [16] Amy Henderson Squillacote and James Ahrens. *The paraview guide*, volume 366. Kitware, 2007.
- [17] David Vanpouille. *Développement de modèles de turbulence adaptés à la simulation des écoulements de convection naturelle à haut nombre de Rayleigh*. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace-ISAIE, 2013.

Annexes

ANNEXE A

CRÉATION DE LA GÉOMÉTRIE AVEC GMSH

Ci-dessous sont décrites les étapes en détail de la création de la géométrie (véhicule et domaine de calcul) avec *Gmsh*.

- Une première étape consiste à créer un fichier 'geo' par exemple "car.geo"(FigureA.1).

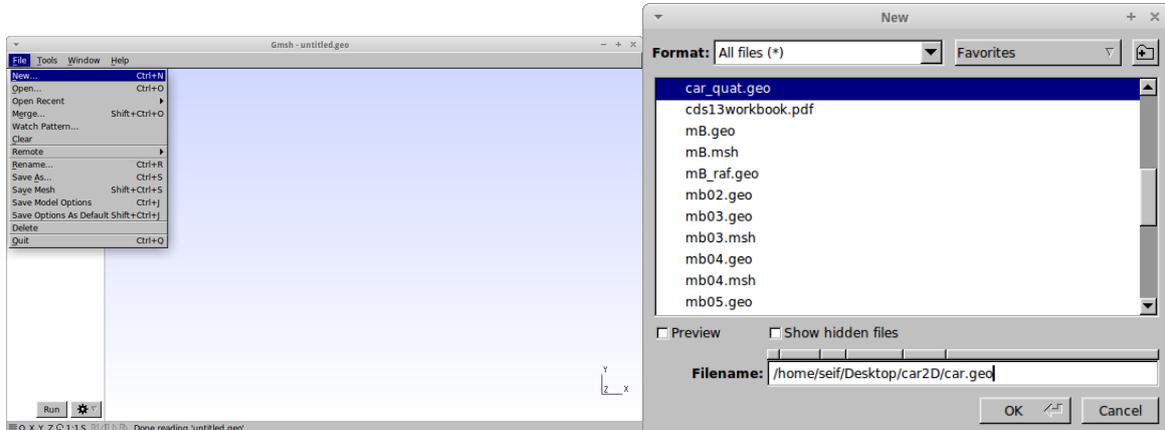


Figure A.1: Création d'un fichier de 'geo'.

Remarque Pour voir la géométrie dans chaque étape, il faut recharger avec le bouton '*Reload*' (FigureA.2) et le bouton '*Edit file*' pour ouvrir et éditer le fichier geo (FigureA.3) dans *Gmsh* GUI :



Figure A.3: Ouvrir le fichier de géométrie.

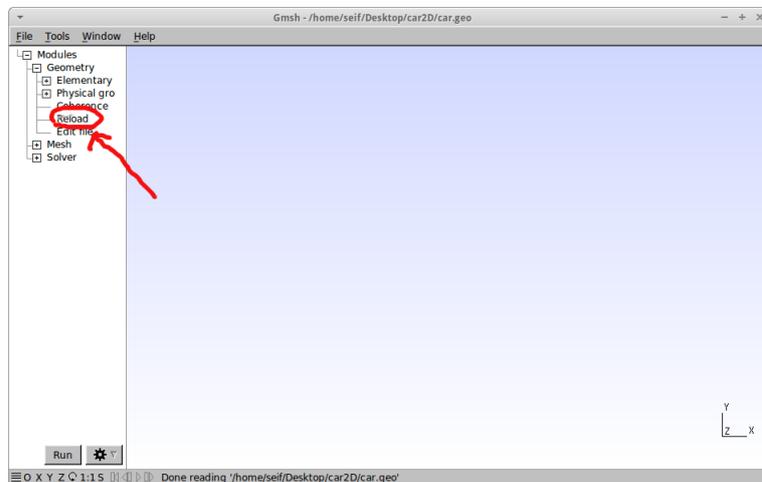


Figure A.2: Recharger et éditer la géométrie.

- Puis on va commencer notre construction dans ce fichier avec la construction la plus simple dans le langage script de *Gmsh* “paramètre”. La commande suivante définit une nouvelle variables ‘lc’ qui représente une longueur caractéristique :

$$lc = 8e-3;$$

Cette variable peut alors être utilisé dans la définition de la plus simple entité élémentaire de *Gmsh* un ‘point’. Un point est défini par une liste de quatre nombres :

1. trois coordonnées (X, Y et Z),
2. une longueur caractéristique (lc) qui définit la taille de l’élément cible au point.

$$\text{Point (1)} = \{0, 14.0028, 0, lc\};$$

La distribution de la taille de l’élément à mailles est alors obtenue par interpolation de ces longueurs caractéristiques à travers la géométrie. Nous pouvons alors définir les autres 118 points du modèle de véhicule avec le même principe ainsi que les coins de

notre domaine (FigureA.4). mais, on définit d’abord, dans l’entête du fichier, toutes les variables dont nous aurons besoin dans les prochaines étapes.

$bx=80e-3$ “longueur caractéristique de les coins de notre domaine”

$L = 297.8168$; $h = 83.77167$; “L la longueur de véhicule, h la hauteur de véhicule+ la garde au sol en [mm] ”

$Bx = 6.66*L$; $By = 0$; $Bz = 0$;

$Cx = -L$; $Cy = 2.6*L+h$; $Cz = 0$;

$Dx = 6.66*L$; $Dy = 2.6*L+h$; $Dz = 0$;

“A, B, C, D sont les coordonnées des points des coins du domaine de calcul ”

$Point(1001) = \{ Ax, Ay, Az, bx \}$;

$Point(1002) = \{ Bx, By, Bz, bx \}$;

$Point(1003) = \{ Cx, Cy, Cz, bx \}$;

$Point(1004) = \{ Dx, Dy, Dz, bx \}$;

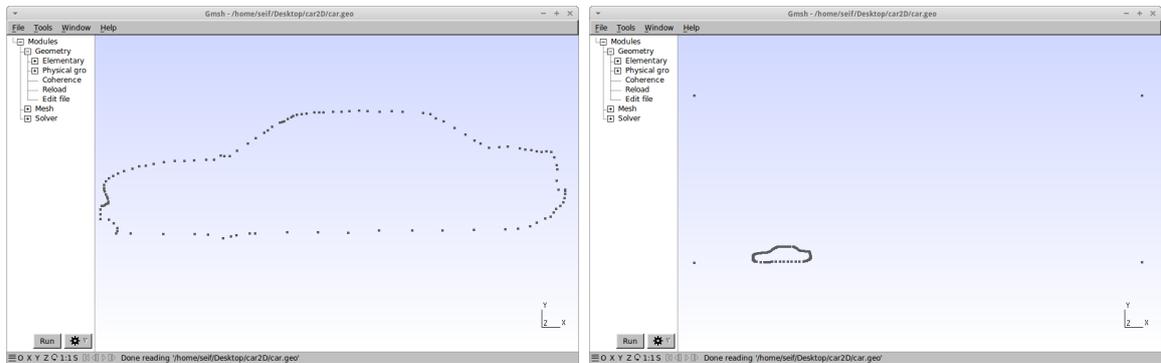


Figure A.4: Création des coordonnées du véhicule et les limites du domaine de calcul.

- Les courbes sont les deuxième type d’entités élémentaires de *Gmsh*, et parmi les courbes on utilise ‘*Line*’ (Une ligne droite) et ‘*Spline*’ (un segment de ligne droite)(FigureA.5).

Bords et limites du domaine :

$Line(1) = \{1001, 1002\}$;

$Line(2) = \{1002, 1004\}$;

$$\text{Line}(3) = \{1004, 1003\};$$

$$\text{Line}(4) = \{1003, 1001\};$$

Bords du véhicule :

$$\text{Spline}(5) = \{1 : 6\};$$

$$\text{Spline}(6) = \{6 : 11\};$$

$$\text{Spline}(7) = \{11 : 35\};$$

$$\text{Line}(8) = \{35, 36\};$$

$$\text{Spline}(9) = \{36 : 117\};$$

$$\text{Spline}(10) = \{117 : 119, 1\};$$

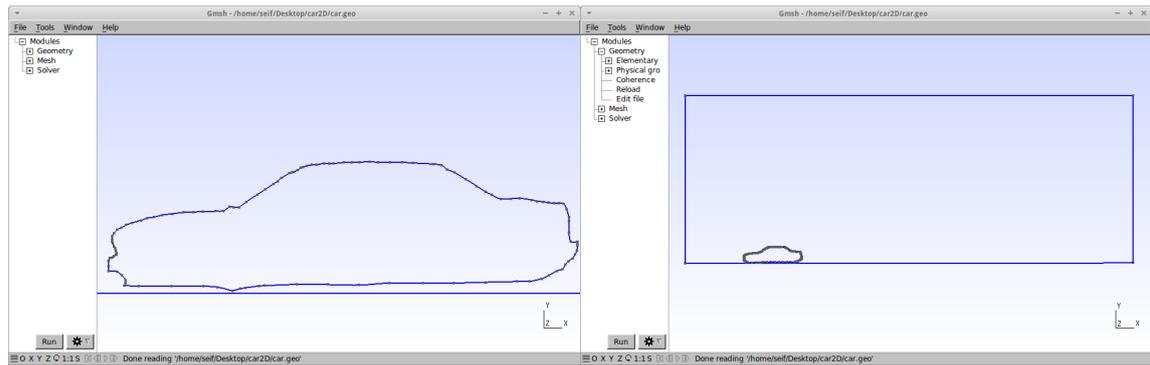


Figure A.5: Bords et limites du véhicule et du domaine de calcul.

- La troisième entité élémentaire est la surface. Afin de définir notre surface (voir figure) à partir des courbes définies ci-dessus un circuit de lignes “*Line loop*” doit d’abord être défini. C’est une liste de lignes connectées, un signe étant associé à chaque ligne (en fonction de l’orientation de la ligne) :

$$\text{LineLoop}(11) = \{1, 2, 3, 4\};$$

$$\text{LineLoop}(12) = \{5 : 10\};$$

On peut alors définir la surface comme une liste de “*Line loop*” :

$$\text{PlaneSurface}(1) = \{11, 12\};$$

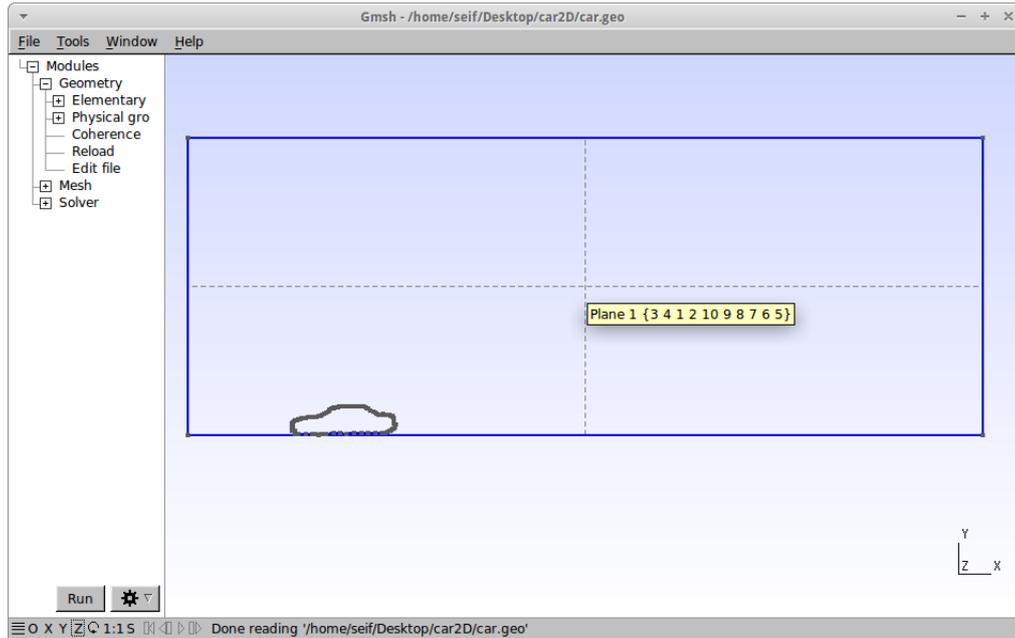


Figure A.6: *Création de la surface.*

- Gmsh offre aussi des outils pour transformer (*'translate', 'rotate', ...etc.*) les entités élémentaires ou des copies d'entités élémentaires. Donc pour convertir de [mm] au [m], on utilise la commande “*Dilate*” avec la structure :

“ *Dilate*{*{expression-liste}, expression}{transformer-liste}*”.

Scale toutes les entités élémentaires de la liste soumise la transformation par une ‘*expression*’ du facteur. L’*expression-liste* doit contenir trois expressions donnant les direction de la transformation homothétique X, Y et Z :

Dilate{*{0,0,0}, 1e-3}{Surface{1};}*

ANNEXE B

DÉTAILS DES MAILLAGES AVEC GMSH

Les détails des fichiers *Gmsh* utilisés pour réaliser les différents maillages sont mentionnés ci-dessous.

B.1 Maillage avec un champ parallélépipédique “Box field”

’Box field’ : spécifie la taille des éléments à l’intérieur et à l’extérieur d’une région du parallélépipède. La valeur de ce champ est *VIn* à l’intérieur du parallélépipède *VOut* à l’extérieur du parallélépipède. Le parallélépipède est donné par :

$$Xmin \leq x \leq XMax$$

$$YMin \leq y \leq YMax$$

$$ZMin \leq z \leq ZMax$$

Le maillage : Nous incluons d’abord le fichier geo précédent “car.geo” dans un autre nouveau fichier geo “Box.geo”, on utilise alors la commande :

```
Include "car.geo";
```

Nous pouvons utiliser “Box field” pour imposer un changement radical dans la taille des éléments dans un rectangle près du véhicule. Le but est de créer un maillage qui fournit des résultats plus précis.

– 1er rectangle :

```
Field[1] = Box;
```

```
Field[1].VIn = lc / 8; // la taille des éléments à l’intérieur est lc/8
```

```
Field[1].VOut = bx/2; // la taille des éléments à l’extérieur est bx/2
```

```

Field[1].XMin = -(L/6)*1e-3;
Field[1].XMax = 1.5*L*1e-3;
Field[1].YMin = 0;
Field[1].YMax = h*1.5*1e-3;
- 2ème rectangle
Field[2] = Box;
Field[2].VIn = lc/4; // la taille des éléments à l'intérieur est cl/4
Field[2].VOut = bx/2; // la taille des éléments à l'extérieur est bx/2 (le même que
le 1er champ)
Field[2].XMin = -(L/3)*1e-3;
Field[2].XMax = 1.75*L*1e-3;
Field[2].YMin = 0;
Field[2].YMax = L*1e-3/1.5;
- 3ème rectangle
Field[3] = Box;
Field[3].VIn = lc/2; // la taille des éléments à l'intérieur est cl/2
Field[3].VOut = bx/2; // la taille des éléments à l'extérieur est bx/2 (le même que
le 1er et le 2ème et champ)
Field[3].XMin = -(L/2)*1e-3;
Field[3].XMax = 2*L*1e-3;
Field[3].YMin = 0;
Field[3].YMax = L*1e-3;
- Finalement, nous allons utiliser le minimum de tous les champs comme champ
de maillage de fond :
Field[4] = Min;
Field[4].FieldsList = {1, 2, 3};
Background Field = 4;

```

Remarque : Vous pouvez également créer des champs directement dans l’interface utilisateur graphique (GUI) en sélectionnant ‘*Define*’ \implies ‘*Fields*’ dans le module ‘*Mesh*’.

Le code OpenFOAM que nous allons utiliser pour nos simulations résoud les problèmes uniquement en 3D par défaut. Il peut être résoudre les problèmes 2d en spécifiant une condition limite vide “*spécial*” sur les limites normales à la 3^{ème} dimension pour laquelle la solution n’est pas nécessaire. Donc et pour ça on utilise la transformation ‘*Extrude*’ (voir figureB.1) avec la structure :

```

“Extrude {expression-liste} { Surface{expression-liste}; Layers{expression-liste};
Recombine; }”,

```

Pour extruder à la fois la géométrie et le maillage des surfaces dans la 2^{ème} expression-liste à l’aide d’une translation, la première expression-liste doit contenir trois expressions donnant X, Y et Z les composantes du vecteur de translation, l’expression-liste de ‘Layers’ donne combien de couches seront créées. L’option recombinaison fera recombinaison les triangles en quadrilatères si possible, pour l’extrusion de surfaces, l’option recombinaison fera recombinaison tétraèdres en prismes, hexaédrique ou pyramides. Donc l’expression utilisée sera :

Extrude {0, 0, 1e-2} {Surface{1}; Layers{1}; Recombine;}

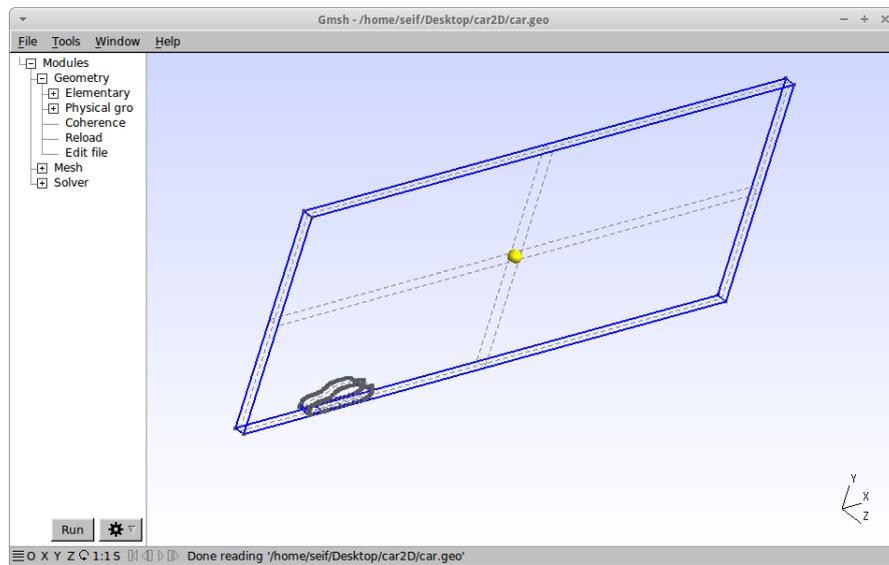
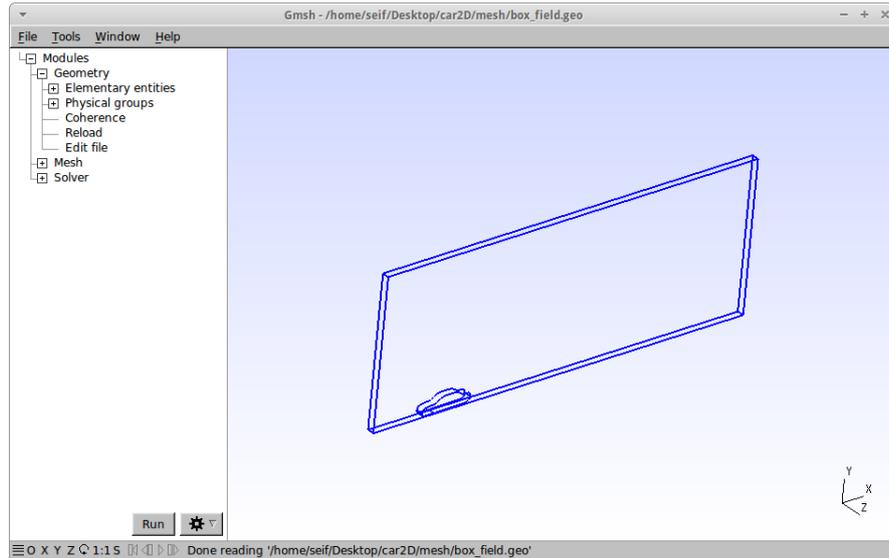


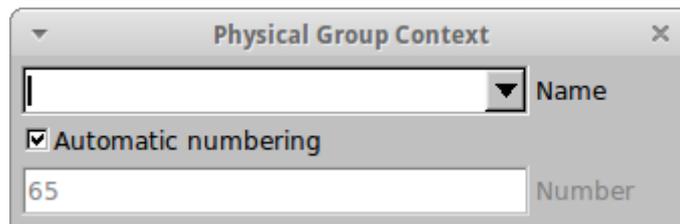
Figure B.1: *Extrusion de la géométrie.*

A ce niveau, *Gmsh* sait tout pour mailler la géométrie. Une seule étape facultative est nécessaire est celles des “entités physiques”. Dans cette étape on utilise l’interface graphique puisqu’elle est la plus facile.

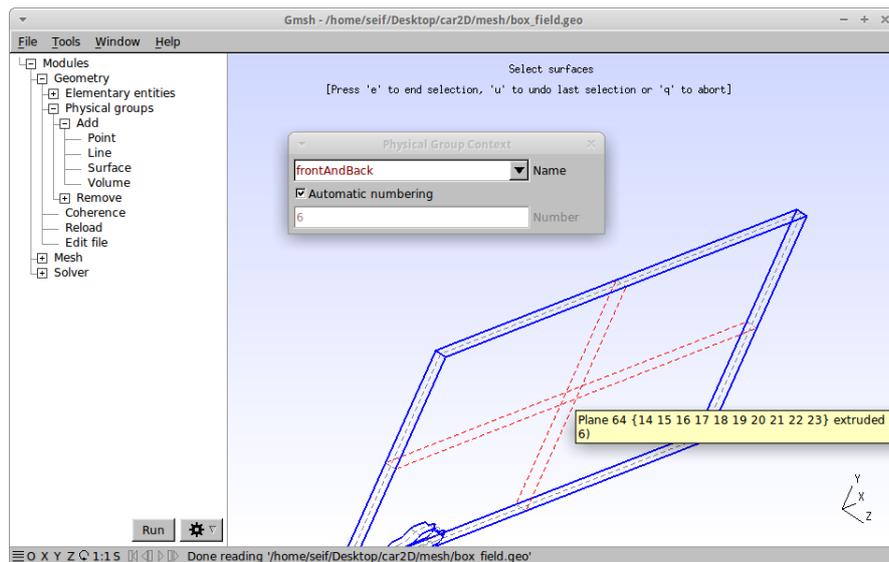
- Appuyez sur le bouton ‘Reload’ dans le menu de “Geometry” pour afficher la géométrie.



- Appuyez sur 'Physical groups' \implies 'add' \implies 'Surface' dans 'Modules Geometry' pour créer une surface physique.



- choisir une surface et nommez-la dans “Physical Groups Context” fenêtre où un numéro d'identification est automatiquement généré.



- En suivant les mêmes étapes, on crée les autres entités physiques (FigureB.2).

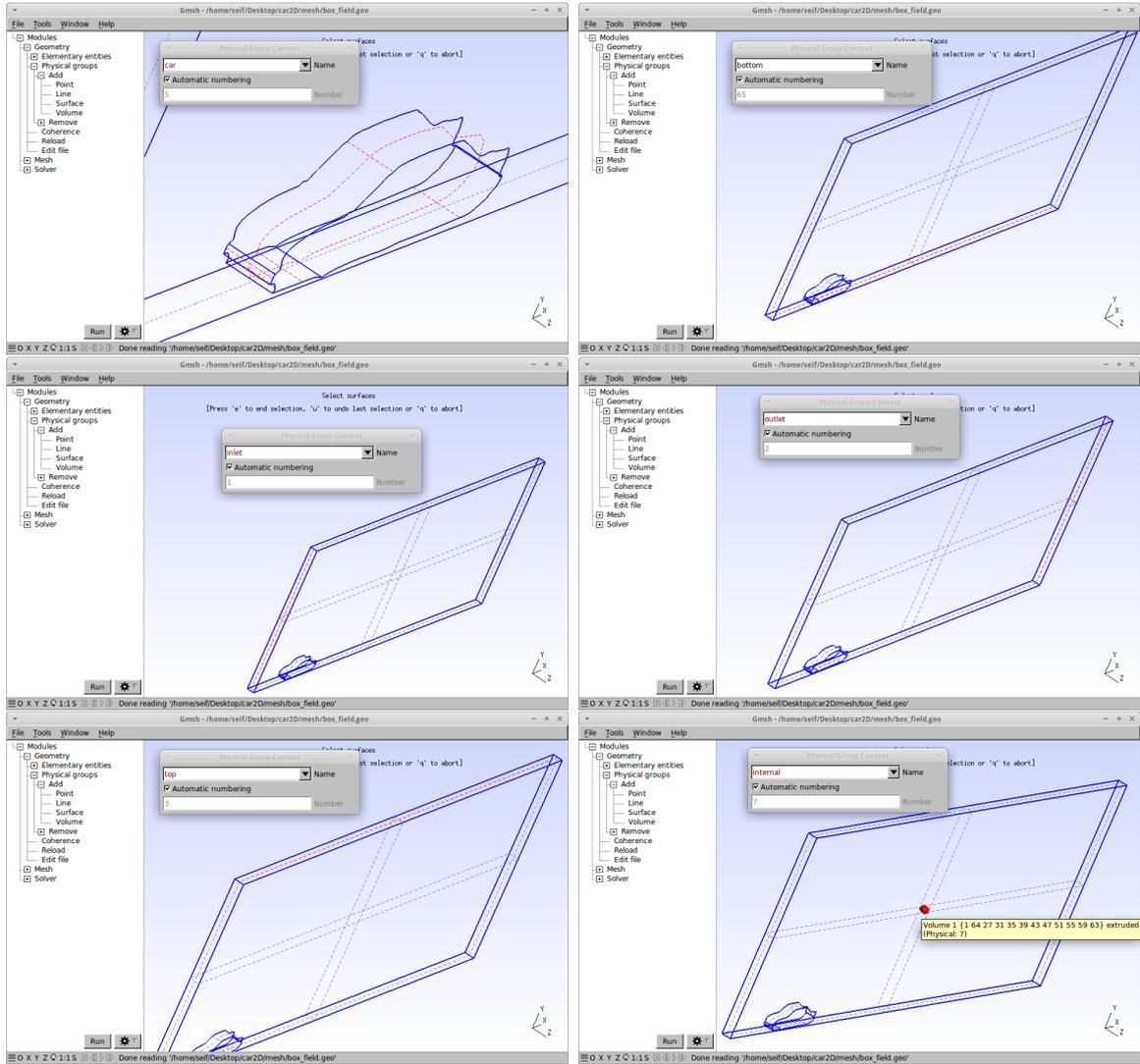


Figure B.2: Création des entités physiques.

Gmsh va enregistrer automatiquement dans le fichier *geo*.

$Physical\ Surface("inlet") = \{31\};$

$Physical\ Surface("outlet") = \{39\};$

$Physical\ Surface("top") = \{27\};$

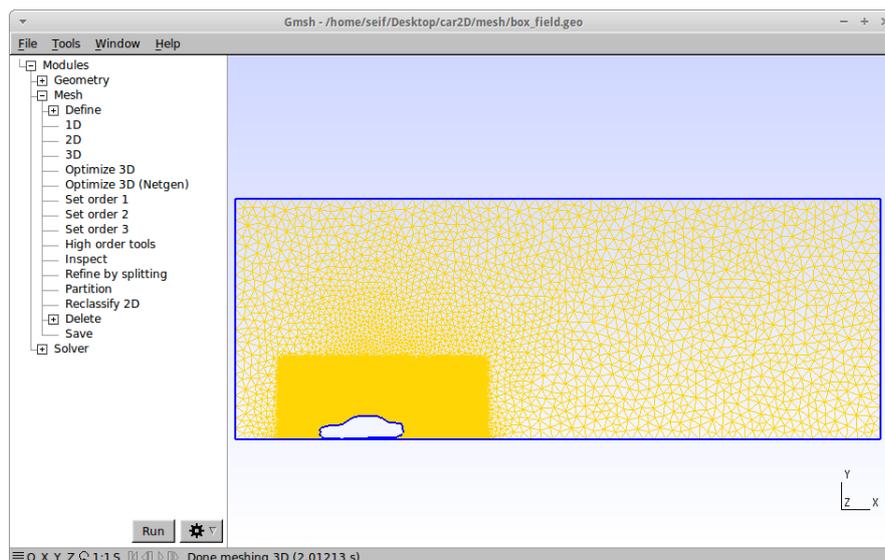
$Physical\ Surface("bottom") = \{35\};$

$Physical\ Surface("car") = \{47, 43, 63, 59, 55, 51\};$

$Physical\ Surface("frontAndBack") = \{64, 1\};$

$Physical\ Volume("internal") = \{1\};$

– Finalement, pour mailler appuyez sur 3D dans 'Modules' \implies 'Mesh' .



- Enregistrez le maillage en sélectionnant ‘Save’ dans le menu ‘Mesh’. Cela va créer un fichier en utilisant le même nom que le fichier .geo, mais avec l’extension ‘.msh’. Ceci sera utilisé par l’utilitaire ‘gmshToFoam’ de OpenFOAM afin de le convertir à son propre format.

B.2 Maillage avec un champ circulaire “Ball field”

A Ball field spécifie la taille des éléments à l’intérieur et à l’extérieur d’une région sphérique. La valeur de ce champ est V_{In} à l’intérieur d’une bille sphérique V_{Out} à l’extérieur. La bille sphérique est défini par :

$$\|dx\|^2 < R^2$$

$$dx = (X - X_C)^2 + (Y - Y_C)^2 + (Z - Z_C)^2$$

$$dx = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2$$

Le maillage

- Ouvrez GMSH et créer un nouveau fichier.
- Ensuite, ouvrez le fichier .geo dans un éditeur de texte.
- Incluez “car.geo” le fichier de notre géométrie dans se nauvaeu fichier .geo avec :
`Include "t1.geo";`
- Comme le dernier on va créez 3 champ près de la véhicule.

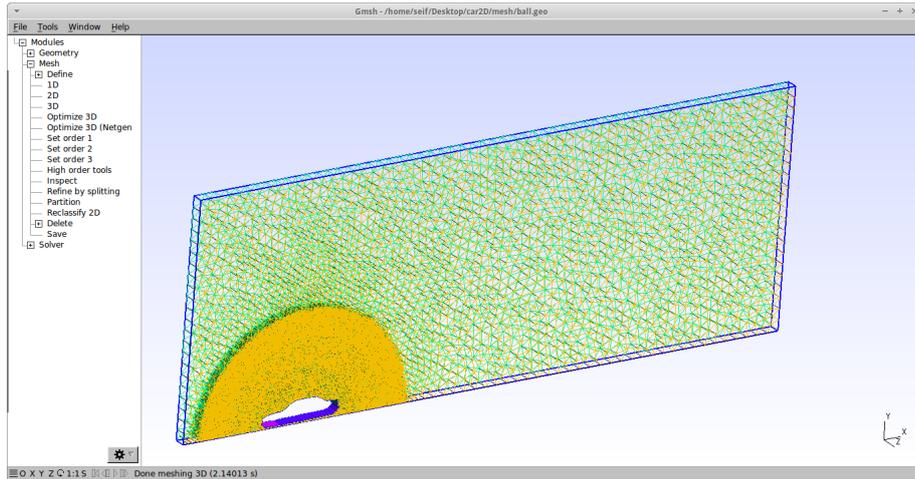
1er champ

```
Field[1] = Ball;
Field[1].Radius = (L/1.5)*m;
Field[1].VIn = lc / 8; Field[1].VOut = bx/2;
Field[1].XCenter = 158.7328*m/*X de point18*/;
```

```

Field[1].YCenter = 0 ; Field[1].ZCenter = 0 ;
2ème champ
Field[2] = Ball ;
Field[2].Radius = (L)*m ;
Field[2].VIn = lc / 4 ;
Field[2].VOut = bx/2 ; // la même VOut comme le 1er champ
Field[2].XCenter = 158.7328*1e-3/*X de point18*/ ;
Field[2].YCenter = 0 ; Field[2].ZCenter = 0 ;
3ème champ
Field[3] = Ball ;
Field[3].Radius = (L/0.75)*1e-3 ;
Field[3].VIn = lc / 2 ;
Field[3].VOut = bx/2 ; // la même VOut comme le 1er et la 2ème champ
Field[3].XCenter = 158.7328*m/*X de point18*/ ;
Field[3].YCenter = 0 ;
Field[3].ZCenter = 0 ;
champ de maillage de fond
Field[4] = Min ;
Field[4].FieldsList = {1, 2, 3} ;
Background Field = 4 ;
- Extrude la géométrie :
Extrude {0, 0, 1e-2} {
  Surface{1} ;
  Layers{1} ;
  Recombine ;
}
- Les entités physiques :
Physical Surface("inlet") = {31} ;
Physical Surface("outlet") = {39} ;
Physical Surface("top") = {27} ;
Physical Surface("btom") = {35} ;
Physical Surface("car") = {47, 43, 63, 59, 55, 51} ;
Physical Surface("frontAndBack") = {64, 1} ;
Physical Volume("internal") = {1} ;
- Mailler la géométrie par sélectionner 3D dans Mesh menu :
- Enregistrez le maillage par sélectionnant "Save" dans le menu Mesh.

```



B.3 Maillage structuré “Multi-Bloc”

Par défaut, *Gmsh* génère des maillages non structurés : des triangles en 2D, des tétraèdres en 3D. Mais il est possible de structurer le maillage. Pour cela, on fait appel à l’algorithme *’transfinite’* 1D, 2D ou 3D pour transformer le maillage d’une ligne, d’une surface ou d’un volume en un maillage triangulaire/tétraédrique plus régulier (Figure B.3a). Par ailleurs, il est possible de générer un maillage quadrangulaire/hexaédrique en appelant également la commande de recombinaison.

La commande *’Recombine Surface’* associée à *’Transfinite’* convertit une fraction des triangles/tétraèdres en quadrangles/hexaèdres (Figure B.3b).

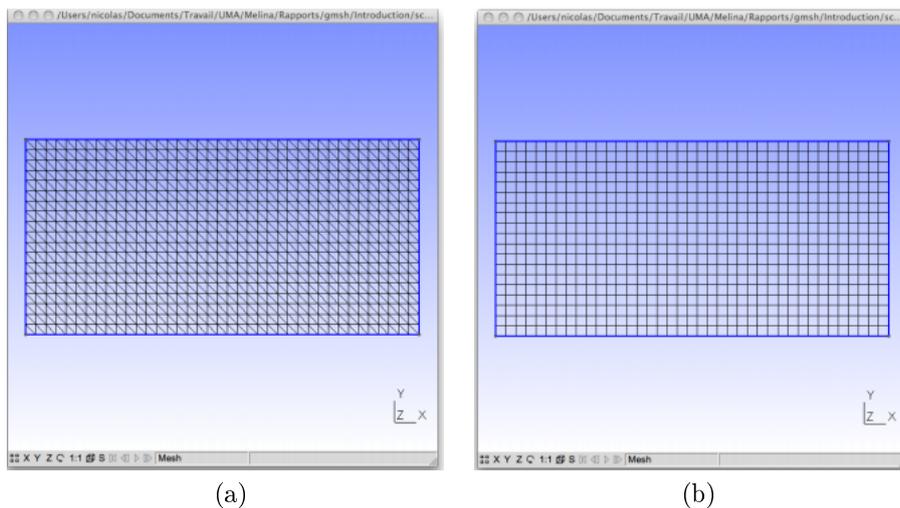


Figure B.3: Visualisation du maillage généré sur un domaine rectangulaire avec *transfinite* (à gauche) et avec *transfinite* et *recombine* (à droite).

- Ouvrez *Gmsh* et créer un nouveau fichier. Ensuite, ouvrez le fichier .geo dans un éditeur de texte, et on crée la géométrie et les blocs (Figure B.4) :

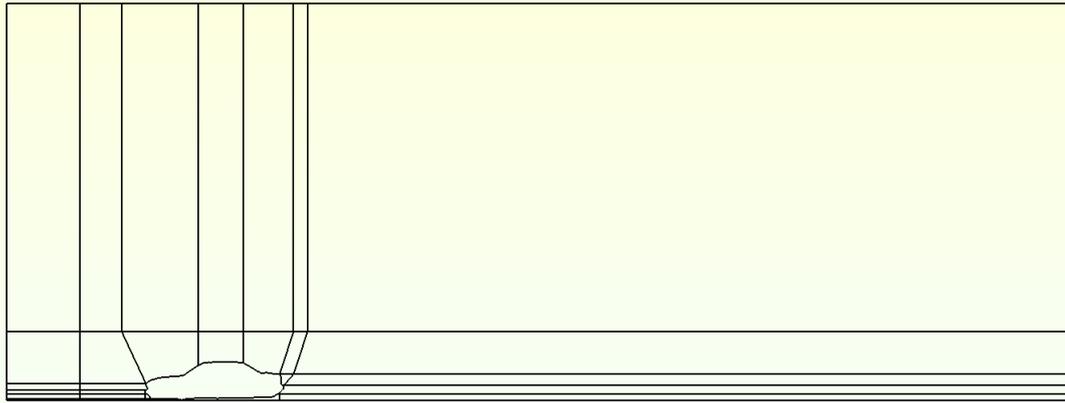
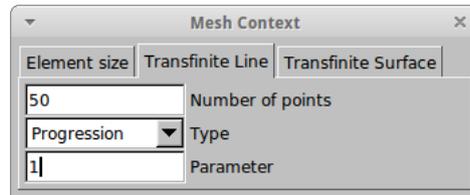


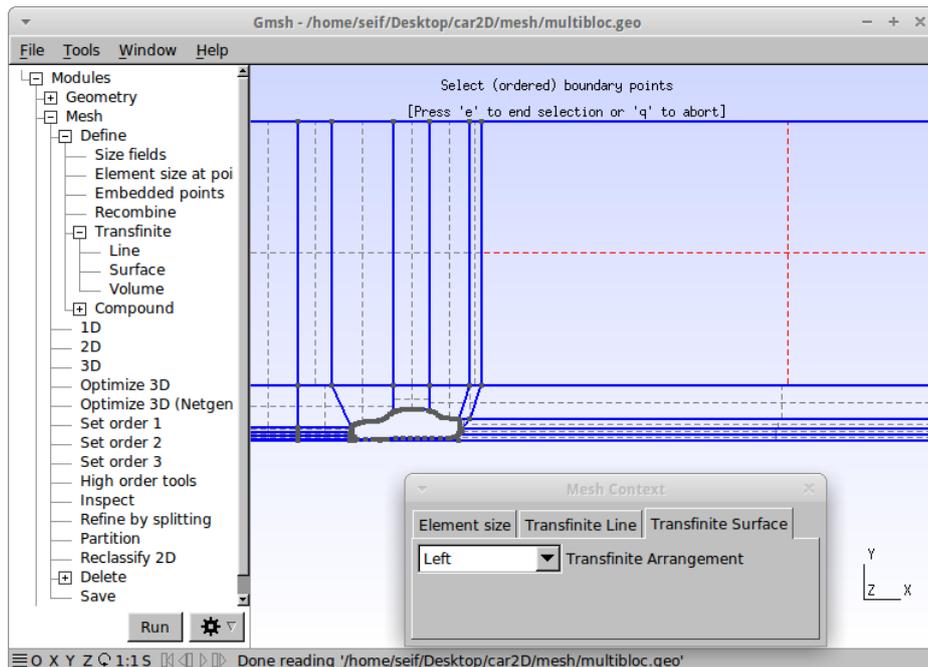
Figure B.4: Décomposition Multibloc du domaine de calcul.

- Dans la 2^{ème} étape, on utilise les commandes 'Transfinite' et 'Recombine Surface' sur la structure du maillage :

Dans Mesh menu sélectionner 'Define' \implies 'Transfinite' \implies 'Line'



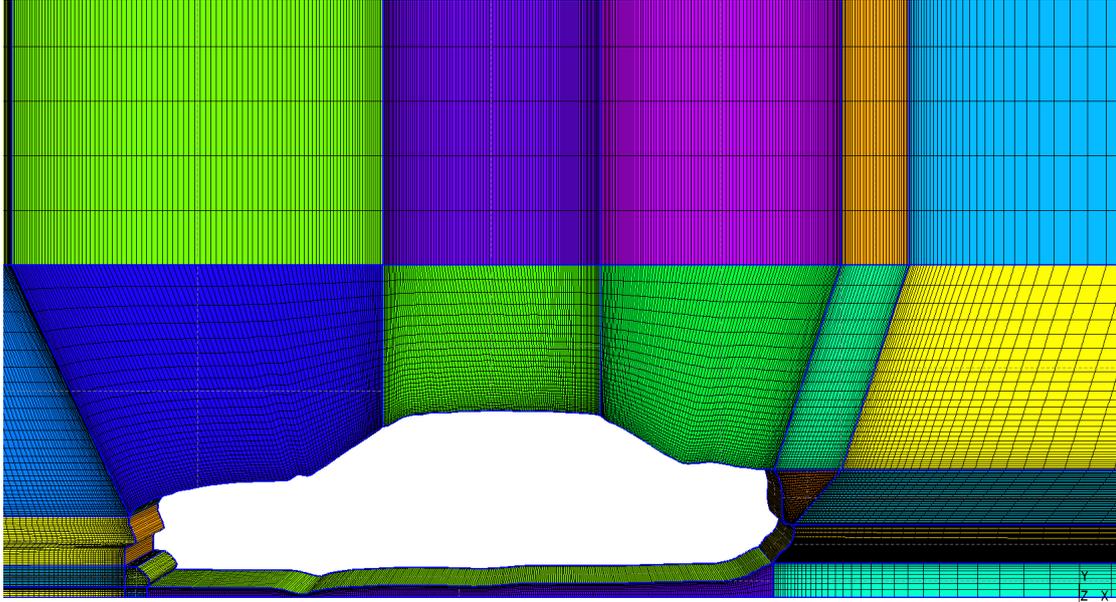
- choisissez le nombre de points dans les lignes choisi. Le type de raffinement 'Progression' ou 'Bump' et leurs paramètres.
- Appuyer sur "e" pour appliquer et mettre fin à la sélection. Après sélectionner 'Transfinite' sur toutes les lignes appuyer sur "q" pour quitter.
- Puis sélectionner 'Transfinite' \implies 'Surface' et sélectionner toutes les surfaces dans la géométrie et appuyer sur "e" pour appliquer puis sur "q" pour quitter.



- Sélectionner 'Define' \implies 'Recombine', choisissez toutes les surface et appuyer sur "e" puis "q" pour quitter.
- Extruder la géométrie.

Extrude {0, 0, 1e-2} {Surface{29}; Layers{1}; Recombine; }

- Définissez les entités physiques .
- Mailler la géométrie en sélectionnant 3D dans 'Mesh Menu' :



B.4 Maillage Hybride

Dans cette section on va essayer de faire un maillage structuré de type couche limite autour du véhicule et un maillage non structuré pour le domaine de calcul loin du véhicule.

- Encore une fois, créez un nouveau fichier .geo dans *Gmsh* et entrez les points qui correspondent aux coordonnées du véhicule ainsi que les limites du domaine d'étude.
- Définissez les point de la couche limite (épaisseur de 3 [mm] en générale) avec sélection de 'Elementary entities' \implies 'Translate' \implies 'Duplicate' point dans le menu de 'Geometry'.

Gmsh saura transformer les doubles des points choisi avec les nouveaux composants donné dans la fenêtre de la figure (B.5).

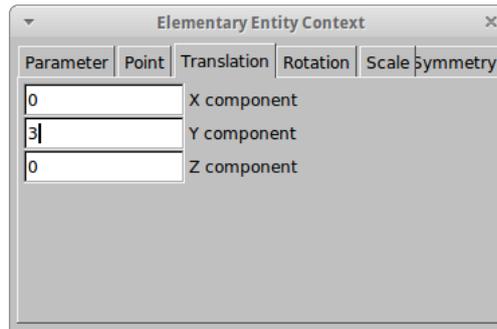
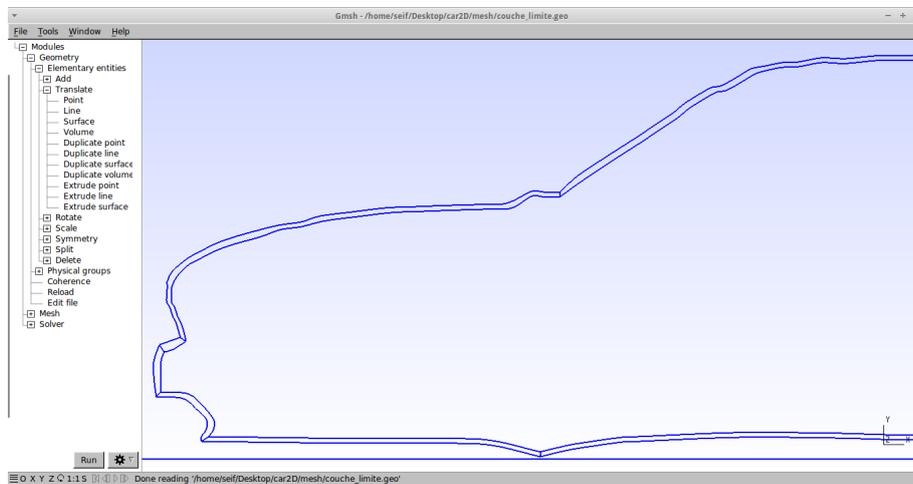


Figure B.5: Fenêtre de transformation.

- Définissez les lignes et les surfaces de la géométrie (diviser la couche limite en blocs pour la structure du maillage).



- Extruder la géométrie et définissez les entités physiques.
- Mailler la couche limite en quadrangulaire avec '*Transfinite*' et '*Recombine*', et créez un champ rectangulaire près du véhicule avec '*Box field*', et appuyer 3D dans le menu '*Mesh*'.

ANNEXE C

DÉTAILS DES SIMULATIONS AVEC OPENFOAM

Le dossier de travail pour lancer la simulation doit être comme dans la figure suivante :

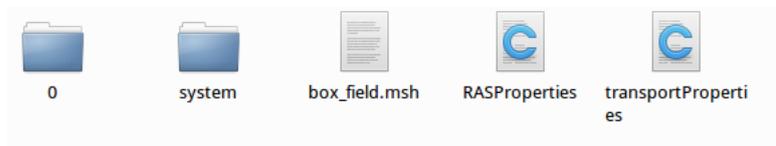
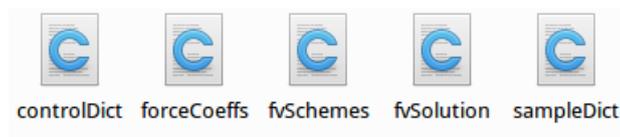


Figure C.1: *Structure du dossier de travail.*

Le dossier temporel "0" contient les valeurs des différentes grandeurs des conditions aux limites.



Le dossier "system" contient les fichiers utilisateurs.



- Le premier pas de notre simulation est de convertir le maillage de format '.msh' et vérifier sa qualité. Pour cela ouvrir un terminal "*OpenFoam Terminal*" et accéder au dossier de travail :

- Convertissez le format .msh avec la ligne de commande “ *gmshToFoam <meshFile.msh>* ”. A ce moment OpenFOAM va créer un nouveau répertoire avec le nom “*constant*” :



et les noms que nous avons donné aux surfaces dans Gmsh sont importées dans OpenFoam :

```

Terminal
File Edit View Terminal Go Help
total:178345
hex :0
prism:178345
pyr :0
tet :0

CellZones:
Zone Size
0 178345

Skipping tag at line 716243
Patch 0 gets name frontAndBack
Patch 1 gets name top
Patch 2 gets name inlet
Patch 3 gets name bttom
Patch 4 gets name outlet
Patch 5 gets name car

--> FOAM Warning :
From function polyMesh::polyMesh(... construct from shapes...)
in file meshes/polyMesh/polyMeshFromShapeMesh.C at line 619
Found 358111 undefined faces in mesh; adding to default patch.
Finding faces of patch 0
Finding faces of patch 1

```

- Pour vérifier la qualité du maillage tapez la ligne de commande *<chekMesh>*

```

Terminal
File Edit View Terminal Go Help
car 683 1366 ok (non-closed singly connected)

Checking geometry...
Overall domain bounding box (-0.297817 0 0) (1.98346 0.858095 0.04)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (9.33722e-19 -1.73343e-17 -3.40677e-23) OK.
Max cell openness = 2.52972e-16 OK.
Max aspect ratio = 217.608 OK.
Minumum face area = 1.59541e-07. Maximum face area = 0.00234889. Face area
magnitudes OK.
Min volume = 6.38162e-09. Max volume = 4.21368e-05. Total volume = 0.077601
6. Cell volumes OK.
Mesh non-orthogonality Max: 46.0758 average: 7.14891
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 0.591716 OK.
Coupled point location match (average 0) OK.

Mesh OK.

End
seif@seif-Inspiron-N5110:~/Desktop/dimoCase/box$

```

- Copier les fichiers “*transportProperties*” et “*RASProperties*” dans la répertoire “*constant*”. Puis modifier les 3 fichiers suivants :
- On est dans une simulation 2D donc il faut changer le fichier “*boundary*” dans *constant* \implies *polyMesh*. Laissez uniquement les patches voulus pour la simulation afin de fonctionner. Modifiez le numéro en haut du fichier texte qui indique le nombre de correctifs ‘*patch*’ qui doivent être mis en place :

```

/*-----* C++ *-----*/
/ ===== /
/ || / Field / OpenFOAM: The Open Source CFD Toolbox /
/ || / Operation / Version: 2.0.1 /
/ || / And / Web: www.OpenFOAM.com /
/ ||/ Manipulation /
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        polyBoundaryMesh;
    location     "constant/polyMesh";
    object       boundary;
}
// ***** //
6
(
    frontAndBack
    {
        type            empty;
        nFaces          71788;
        startFace       70780;
    }
    bottom
    {
        type            wall;
        nFaces          465;
        startFace       142568;
    }
    inlet
    {
        type            patch;
        nFaces          99;
        startFace       143033;
    }
    top
    {
        type            symmetryPlane;
        nFaces          453;
        startFace       143132;
    }
    car
    {
        type            wall;
        nFaces          829;
        startFace       143585;
    }
    outlet
    {
        type            patch;
        nFaces          170;
        startFace       144414;
    }
}
// ***** //

```

– Dans le dossier 0 changer le fichier de pression “p” et de vitesse “U” comme ci-dessous :

fichier "p"

```
/*-----* C++ *-----*/
/ ===== /
/ || / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
/ || / O p e r a t i o n / V e r s i o n : 2.0.1 /
/ || / A n d / W e b : w w w . O p e n F O A M . c o m /
/ || / M a n i p u l a t i o n /
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       p;
}
// ***** //
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    frontAndBack
    {
        type          empty;
    }
    bottom
    {
        type          zeroGradient;
    }
    inlet
    {
        type          freestreamPressure;
    }
    top
    {
        type          symmetryPlane;
    }
    car
    {
        type          zeroGradient;
    }
    outlet
    {
        type          freestreamPressure;
    }
}
// ***** //
```

fichier "U" :

```
/*-----* C++ *-----*/
/ ===== /
/ || / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
/ || / O p e r a t i o n / V e r s i o n : 2.0.1 /
/ || / A n d / W e b : w w w . O p e n F O A M . c o m /
/ || / M a n i p u l a t i o n /
/*-----*/
FoamFile
{
    version      2.0;
```

8. Pour le tracé de courbes spécifiques lancer la commande `<sample -latestTime>`
Le répertoire de travail sera alors enrichi de la manière suivante :

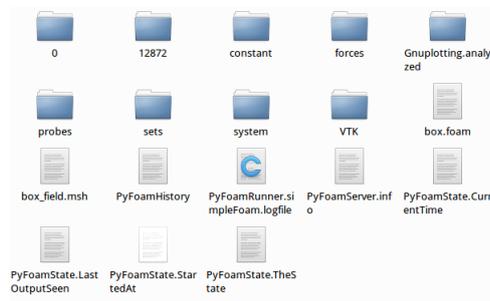


Figure C.2: *Répertoire de travail après calculs.*