



HAL
open science

Optimisation de forme avec la méthode adjointe appliquée aux équations de Lattice-Boltzmann en aérodynamique

Isabelle Cheylan

► **To cite this version:**

Isabelle Cheylan. Optimisation de forme avec la méthode adjointe appliquée aux équations de Lattice-Boltzmann en aérodynamique. Mécanique des fluides [physics.class-ph]. Aix-Marseille Université, 2019. Français. NNT: . tel-02477455

HAL Id: tel-02477455

<https://theses.hal.science/tel-02477455>

Submitted on 13 Feb 2020

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

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

AIX-MARSEILLE UNIVERSITÉ
RENAULT
ECOLE DOCTORALE 353
LABORATOIRE M2P2 : Mécanique, Modélisation et Procédés
Propres

Thèse présentée pour obtenir le grade universitaire de docteur

Mention/Spécialité : Mécanique et Physique des Fluides

Isabelle CHEYLAN

Optimisation de forme avec la méthode adjointe appliquée aux équations de Lattice-Boltzmann en aérodynamique

Shape optimization using the adjoint Lattice-Boltzmann method for
aerodynamic applications

Soutenue le 21/03/2019 devant le jury composé de :

François DUBOIS	Conservatoire National des Arts et Métiers	Rapporteur
Christophe CORRE	Ecole Centrale de Lyon	Rapporteur
Grégoire ALLAIRE	Ecole Polytechnique	Examinateur
Maria Vittoria SALVETTI	University of Pisa	Examinatrice
Julien FAVIER	Université d'Aix-Marseille	Examinateur
Denis RICOT	Renault	Examinateur
Pierre SGAUT	Université d'Aix-Marseille	Directeur de thèse
Guillaume FRITZ	Renault	Encadrant

Numéro national de thèse/suffixe local : 2019AIXM0119/007ED353

Remerciements

Mes remerciements vont en premier lieu à mon directeur de thèse Pierre Sagaut, qui a su me guider de façon avisée pendant ces trois années de thèse. Ses conseils toujours justes et ses remarques constructives m'ont été très utiles, et son accueil chaleureux ainsi que nos discussions instructives et sympathiques lors de mes venues dans la cité phocéenne m'ont permis de me sentir immédiatement la bienvenue. Je le remercie pour la confiance qu'il m'a accordée, et j'aimerais lui exprimer à travers ces quelques mots la profonde gratitude et reconnaissance que je lui porte.

J'aimerais aussi remercier Denis Ricot, qui m'a introduite à la méthode de Boltzmann sur réseau, pour sa rigueur, son esprit critique et son encadrement efficace. Malgré son emploi du temps très chargé, je le remercie pour le temps qu'il m'a accordé afin de m'aider pendant ces trois ans. Je tiens également à exprimer mes remerciements à Guillaume Fritz, pour son rôle d'encadrant et pour sa patience. Sa lecture méticuleuse du mémoire de thèse, ainsi que ses conseils lors de la préparation de présentations, m'ont été d'une grande aide. J'aimerais également exprimer ma reconnaissance à Félix Gendre, pour nos discussions enrichissantes sur divers aspects de la méthode de Boltzmann sur réseau. Je remercie aussi les membres de la DEA-TDS4 pour leur accueil chez Renault. Enfin, je remercie Marc Albertelli dont les discussions sur la méthode adjointe et sur les méthodes d'optimisation de façon générale m'ont permis de me former en début de thèse. Je tiens aussi à remercier Sylvain Maillot pour sa gentillesse, sa générosité, et sa bonne humeur quotidiennes qui ont rendu ces trois années de thèse très agréables.

Je remercie également François Dubois et Christophe Corre pour leurs rapports sur ce mémoire qui ont permis de soulever un certain nombre de points intéressants et de perspectives. J'exprime aussi mes remerciements à Grégoire Allaire pour l'intérêt qu'il a porté à mon travail et pour avoir accepté de présider le jury de thèse, ainsi que tous les membres du jury qui ont accepté de juger mon travail.

Enfin, je tiens à remercier mes amis et mes parents qui m'ont soutenue et encouragée pendant ces trois années. Un merci particulier à ma mère qui m'a encouragée à poursuivre cette thèse, et à Anne, Emilie, Audrey et les autres qui se reconnaîtront ...

Pour finir, je remercie Alexandre pour son soutien infaillible, sa patience, et son optimisme quotidien.

Résumé

Cette thèse a pour objectif le développement d'un solveur adjoint dans ProLB, le logiciel d'aérodynamique basé sur la méthode de Lattice-Boltzmann utilisé chez Renault. Ce solveur adjoint permet de calculer les sensibilités surfaciques des efforts aérodynamiques sur un obstacle, typiquement un véhicule, par rapport à la forme de celui-ci. L'objectif final est de le déformer, par des techniques de morphing basées sur une méthode de descente du gradient à pas constant, afin de réduire sa traînée aérodynamique. Dans un premier temps, l'étude de cas 2D laminaires permet de détailler le développement du solveur adjoint étape par étape. Les équations du problème direct dans le cœur du fluide (modèle de collision BGK), en proche paroi (algorithme de Ginzburg), au niveau des raffinements de maillage et des conditions limites sont écrites puis dérivées afin d'en obtenir leur équivalent adjoint. Le choix de la formulation de la force de traînée est un point important car il a un impact direct sur la complexité des développements nécessaires à l'obtention des équations adjointes et du calcul du gradient. Une étude a montré qu'il est plus judicieux de calculer cette force dans le sillage de l'objet plutôt que sur l'objet lui-même comme cela est fait classiquement. L'objectif étant de minimiser la force de traînée moyenne, il a été montré que le meilleur compromis entre l'effort de calcul et la précision des gradients est obtenu en moyennant le champ direct instationnaire sur un nombre de pas de temps suffisant après convergence. Dans un second temps, l'étude de cas 3D turbulents à grandes échelles a permis de montrer que les algorithmes fonctionnant sur des cas 2D laminaires ne sont pas suffisamment stables pour être utilisés dans ce contexte beaucoup plus complexe. Des modifications ont donc été apportées au solveur adjoint afin de pouvoir l'utiliser dans un contexte industriel. Après avoir justifié et documenté les différentes hypothèses retenues, le solveur adjoint a été développé et testé sur un cas test industriel. Il permet d'obtenir une cartographie des sensibilités sur tout le véhicule dans un écoulement à Reynolds élevé. Une boucle d'optimisation complète a été réalisée, avec une étape de lissage des sensibilités, et a permis de réduire la force de traînée de 5%.

Abstract

This work aims at developing an adjoint solver in ProLB, the aerodynamic software based on the Lattice-Boltzmann method used by Renault. The adjoint solver makes it possible to calculate the surface sensitivities of the aerodynamic forces acting on an obstacle, such as a vehicle, with respect to its shape. The final purpose is to deform it, using morphing techniques based on a fixed step gradient descent method, in order to reduce its drag. First, the step by step development process of the adjoint solver is shown through 2D laminar test cases. The equations of the direct problem are written at the core of the fluid (BGK collision model), at the interface between solids and fluids (Ginzburg interpolation), at the grid refinements and at the boundary conditions. They are then derived to find their adjoint formulation. The choice of the drag force expression is important because it has an impact on the complexity of the adjoint equations and on the gradient calculation. It is shown that calculating the drag force in the wake of the obstacle is more adequate than calculating it on the obstacle directly. The aim being to minimize the time-averaged drag force, it is demonstrated that the best trade-off between the gradients accuracy and the computation cost is obtained by averaging the unsteady direct field on a sufficient number of time steps after convergence has been reached. Then, the study of 3D large-scale turbulent cases shows that the algorithms used for the 2D laminar cases are not stable enough to be used in this more complicated context. Changes have therefore been brought to the adjoint solver, in order to use it in an industrial context. Every assumption used for the development of the adjoint solver is justified and referenced. The adjoint solver is finally applied to an industrial test case. It gives a sensitivity map on a vehicle in a high Reynolds number flow. A complete optimization loop is performed, using a smoothing step on the sensitivities, and gives a 5% reduction of the drag force.

Table des matières

Introduction	5
I Bibliographie, stratégies, et problématiques	8
1 Choix d'une méthode d'optimisation	10
1.1 Méthodes d'optimisation	10
1.1.1 Optimisation Paramétrique	11
1.1.2 Optimisation Surfaccique par morphing direct	15
1.1.3 Optimisation Topologique	16
1.2 Boucle d'optimisation complète	19
1.2.1 Fonctionnelle de coût I_0	20
1.2.2 Estimation du gradient	25
1.2.3 Méthode itérative	30
2 Application à la mécanique des fluides	33
2.1 Optimisation avec les équations de Navier-Stokes	33
2.1.1 Equations directes	34
2.1.2 Equations adjointes	34
2.1.3 Calcul du gradient	36
2.2 Optimisation avec les équations de Lattice-Boltzmann	37
2.2.1 Présentation générale de la LBM	37
2.2.2 Etude bibliographique de la méthode adjointe en LBM	39
2.2.3 Equations directes	44
2.2.4 Equations adjointes	45
2.2.5 Calcul du gradient	47
3 Application à un problème industriel	48
3.1 Modèles de collision	48
3.1.1 Opérateur de collision BGK	48
3.1.2 Opérateur de collision DRT	48
3.2 Conditions limites de paroi	49
3.2.1 Condition de rebond : Bounce-Back	50

3.2.2	Interpolation de Ginzburg	51
3.2.3	Méthode de reconstruction utilisée dans ProLB	51
3.3	Raffinement de maillage	52
3.4	Modèle de turbulence	54
3.5	Approximation temporelle du problème direct	56
3.5.1	Direct stationnaire-Adjoint stationnaire	56
3.5.2	Direct quasi instationnaire - Adjoint stationnaire	56
3.5.3	Direct instationnaire - Adjoint instationnaire	57
3.5.4	Comparaison des méthodes	57
Conclusion de la partie I		59
 II Développement des algorithmes LBM adjoints et validation sur des cas 2D académiques		61
 4 Choix de l’algorithme de paroi et des raffinements de maillage		64
4.1	Etude préliminaire	64
4.1.1	Contrôle actif en acoustique : diminution du bruit	64
4.1.2	Résultats	67
4.2	Condition de rebond : Bounce-back	72
4.2.1	Contrôle actif en acoustique en présence d’un obstacle : diminution du bruit	72
4.2.2	Résultats	74
4.3	Méthode de reconstruction utilisée dans ProLB	76
4.3.1	Contrôle actif en acoustique en présence d’un obstacle : diminution du bruit	76
4.3.2	Résultats	77
4.4	Interpolation de Ginzburg	78
4.4.1	Contrôle actif en acoustique en présence d’un obstacle : diminution des efforts sur l’obstacle	78
4.4.2	Résultats	81
4.5	Raffinements de maillage	83
4.5.1	Algorithmes du problème direct	83
4.5.2	Algorithmes du problème adjoint	84
4.5.3	Résultats	85
4.6	Conclusion	87
 5 Choix de la fonctionnelle de coût		89
5.1	Force de traînée : Intégration sur l’obstacle	89
5.1.1	Ecoulement 2D stationnaire autour un cylindre	89
5.1.2	Résultats	91
5.2	Force de traînée : Intégration sur un volume de contrôle	93
5.2.1	Ecoulement 2D stationnaire autour d’un cylindre	93

5.2.2	Résultats	96
5.3	Conclusion	102
6	Choix de l'approximation temporelle	103
6.1	Direct instationnaire - Adjoint instationnaire	103
6.1.1	Ecoulement 2D instationnaire autour d'un cylindre	103
6.1.2	Résultats	105
6.2	Direct instationnaire moyenné - Adjoint stationnaire	107
6.2.1	Ecoulement 2D instationnaire moyenné autour d'un cylindre	107
6.2.2	Résultats	108
6.3	Conclusion	110
	Conclusion de la partie II	111
	III Extension des algorithmes adjoints pour les applications 3D industrielles	113
7	Résultats	115
7.1	Conclusion	121
	Conclusion de la partie III	122
	Conclusion générale	124

Nomenclature

α	=	paramètre d'optimisation
\mathbf{c}	=	vitesse adimensionnelles
c_s	=	vitesse du son ($m.s^{-1}$)
C_S	=	constante de Smagorinsky
d	=	distance à la paroi (m)
δ	=	symbole de Kronecker
f	=	fonction de distribution ($kg.m^{-3}$)
f^{eq}	=	fonction d'équilibre ($kg.m^{-3}$)
f^{col}	=	fonction de distribution post-collision ($kg.m^{-3}$)
f^*	=	fonction de distribution adjointe
$f^{*,eq}$	=	fonction d'équilibre adjointe
$J_{LBB}^{*,eq}$	=	fonction d'équilibre adjointe pour le link bounce-back
f^f	=	fonction de distribution de la grille fine ($kg.m^{-3}$)
f^c	=	fonction de distribution de la grille grossière ($kg.m^{-3}$)
\mathbf{F}	=	force aérodynamique ($kg.m.s^{-2}$)
F_x	=	force de traînée ($kg.m.s^{-2}$)
$\mathcal{H}^{(2)}$	=	polynôme de Hermite du second ordre ($m^2.s^{-2}$)
I_0	=	fonction coût
L	=	Lagrangien
\mathbf{j}	=	quantité de mouvement ($kg.m^{-2}.s^{-1}$)
\mathbf{j}^*	=	quantité de mouvement adjointe
\mathbf{J}_{LBB}^*	=	quantité de mouvement adjointe pour le link bounce-back
ν	=	viscosité moléculaire ($m^2.s^{-1}$)
ν_t	=	viscosité turbulente ($m^2.s^{-1}$)
ρ	=	densité ($kg.m^{-3}$)
ρ^*	=	densité adjointe
ρ_{LBB}^*	=	densité adjointe pour le link bounce-back
p	=	pression statique ($kg.m^{-1}.s^{-2}$)
p_0	=	pression statique de référence ($kg.m^{-1}.s^{-2}$)
$P^{(1)}$	=	partie hors-équilibre du moment de second ordre ($kg.m^{-1}.s^{-2}$)
R	=	équations d'état
τ	=	tenseur des contraintes visqueuses ($kg.m^{-1}.s^{-2}$)
τ^N	=	temps de relaxation numérique sans dimension
τ^S	=	temps de relaxation physique sans dimension
τ^f	=	temps de relaxation de la grille fine
τ^c	=	temps de relaxation de la grille grossière
\mathbf{u}_0	=	vitesse de référence ($m.s^{-1}$)
\mathbf{u}	=	vecteur vitesse ($m.s^{-1}$)
u_x, u_y, u_z	=	composantes de la vitesse ($m.s^{-1}$)
ω	=	coefficient de pondération

Introduction

Dans le monde automobile, les plannings de développement de nouveaux véhicules tendent à être de plus en plus courts, et les normes en termes d'efficacité énergétique de plus en plus strictes. Le déroulement d'un projet de développement concernant l'aérodynamique globale d'un véhicule chez Renault suit le schéma 1. Nous pouvons voir sur ce schéma que l'aérodynamique du véhicule est découpée en trois parts chez Renault : celle due à l'architecture (c'est-à-dire les grandes côtes du véhicule), celle due au design (la forme de l'épaule ou de l'arrière de véhicule par exemple), et la part due aux technologies que l'on ajoute au véhicule (carénage ou volets pilotés par exemple).

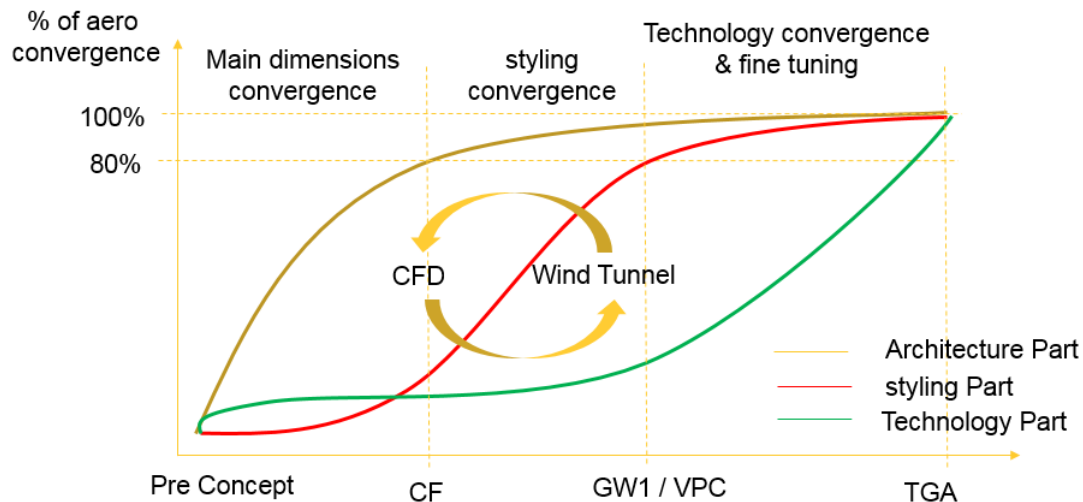


FIGURE 1 – Déroulement d'un projet de développement d'un véhicule pour l'aérodynamique du cadre initial (Pre Concept) à fin du développement (TGA)

Dans ce contexte, les designers échangent régulièrement avec les ingénieurs pour dessiner un véhicule à la fois esthétique et performant en terme d'aérodynamique. Ces échanges prennent place en phase de convergence design d'un nouveau véhicule, entre le jalon CF (figeage des grandes côtes) et le VPC (figeage du style), très souvent sur des maquettes à l'échelle 2/5 faite en claie. Les designers proposent plusieurs designs de véhicule en début de projet, dont un seul sera retenu pour la suite des développements.

Dans le processus actuel d'optimisation de forme aérodynamique des véhicules, la discussion entre le département du design et celui des ingénieurs aérodynamiciens se fait via des règles métiers, basées sur l'expérience et de très nombreux essais en soufflerie. Ces règles métiers concernent par exemple la resserrée de custode, la fin de pavillon, ou encore la largeur de flanc de roue arrière.

Une autre technique d'optimisation de forme aérodynamique utilisée chez Renault est basée sur des plans d'expériences sur certaines pièces du véhicule. Ce type d'optimisation est réalisé plus tardivement au cours du développement, une fois qu'un seul design a été retenu. L'optimisation par plan d'expériences n'est pas réalisée sur la forme globale du véhicule mais sur des pièces spécifiques, telles que le soubassement, le rétroviseur ou les jantes de pneu par exemple. Cela permet d'optimiser par parties ou plutôt par pièces le véhicule, mais le nombre de paramètres serait trop grand pour optimiser la forme globale du véhicule avec cette technique.

Renault souhaite ainsi se doter d'une nouvelle méthodologie numérique opérationnelle permettant de faire de l'optimisation en aérodynamique sur la forme globale du véhicule. La méthode adjointe est bien adaptée pour ce type de problèmes, et permettrait de faciliter les échanges entre designers et ingénieurs puisqu'elle est basée sur un outil objectif fondé sur des équations mathématiques. De plus il est possible d'obtenir grâce à cette technique une cartographie sur tout le véhicule des sensibilités de la force de traînée par rapport à sa forme. Cela permettrait donc aux ingénieurs de proposer aux designers des idées pour guider la forme de la voiture vers un résultat optimisé. Nous pouvons imaginer utiliser la méthode adjointe sur la forme globale du véhicule lors de la phase de convergence design, afin de faciliter les discussions entre designers et aérodynamiciens.

Plusieurs boucles de calculs sont réalisées au cours du projet, afin de déterminer le coefficient aérodynamique du véhicule et de le comparer aux résultats des essais en soufflerie. Chez Renault, l'aérodynamique externe du véhicule est simulée avec les équations de Lattice-Boltzmann grâce au logiciel ProLB. L'objectif est par conséquent d'appliquer la méthode adjointe aux équations de Lattice-Boltzmann, ce qui représente une nouveauté puisqu'il n'existe pas de solveur adjoint industriel pour les équations de la LBM (Lattice-Boltzmann Method) à la connaissance de l'auteur.

Dans le cadre de cette thèse, nous nous intéressons surtout à la réduction de la traînée qui a plusieurs effets liés sur un véhicule : réduction de la consommation, capacité d'accélération améliorée et vitesse maximale augmentée. Les pièces visées sont toutes celles composant la forme externe du véhicule, du rétroviseur au becquet arrière en passant par les montants de baie. De plus, l'essor des véhicules hybrides ainsi que des véhicules électriques pour lesquels une consommation minimale est souhaitée justifie pleinement ce sujet de recherches. Le recours à l'optimisation par la méthode adjointe permettrait d'obtenir des gains très intéressants sur le SCx et donc sur la consommation énergétique puisqu'à partir d'une réduction de quelques millièmes de SCx, les gains en émission de

CO2 sont importants.

L'objectif de la thèse est de fournir un processus d'optimisation numérique par méthode adjointe appliquée aux équations de Lattice-Boltzmann. Cela permettra à terme de modifier la forme du véhicule grâce à des outils dédiés reposant sur un solveur d'optimisation, pour améliorer les performances aérodynamiques du véhicule et donc sa consommation énergétique.

La première partie de cette thèse regroupe le premier chapitre qui concerne la bibliographie réalisée sur les méthodes d'optimisation de façon générale. Nous verrons ensuite dans le second chapitre comment appliquer ces méthodes d'optimisation en mécanique des fluides, et enfin dans le troisième chapitre leur utilisation avec un outil industriel et toutes les complexités que cela amène. La seconde partie montre les différentes hypothèses étudiées pour le développement d'un solveur adjoint sur des cas tests 2D académiques. Enfin la dernière partie se concentre sur le développement du solveur adjoint sur des cas plus complexes qui se rapprochent d'une utilisation industrielle. Nous verrons dans cette dernière partie les complexités qu'un cas très turbulent à grandes échelles apporte, et comment nous avons mis en place des stratégies pour les surmonter.

Première partie

**Bibliographie, stratégies, et
problématiques**

Afin d'expliquer le positionnement du doctorat et les choix effectués au cours de celui-ci, la partie bibliographique va suivre le schéma de la figure 2.

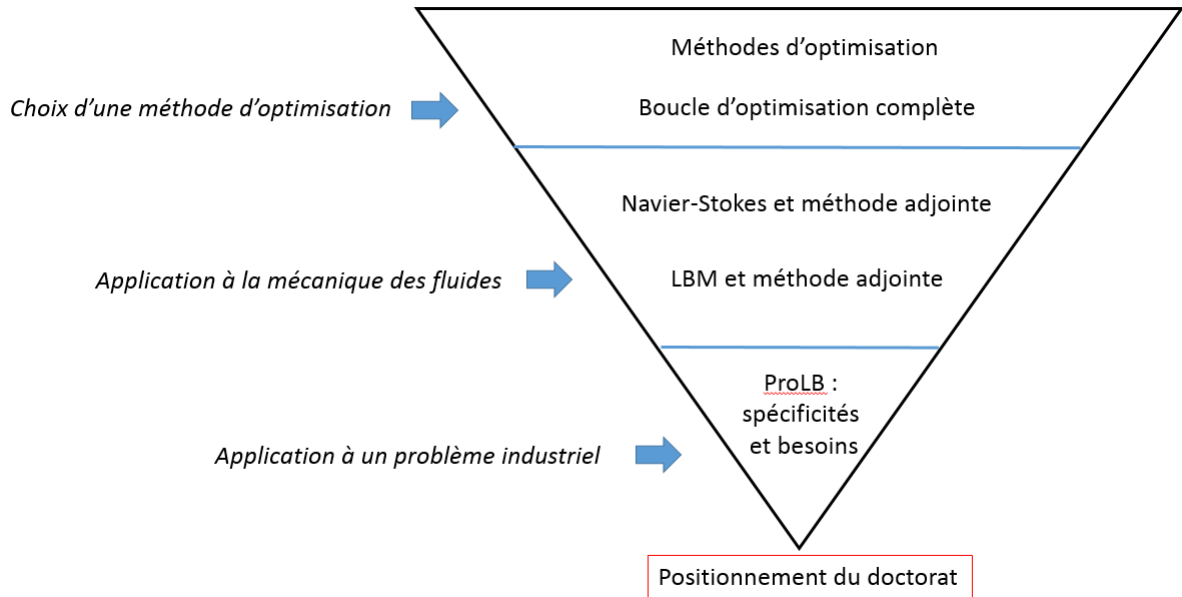


FIGURE 2 – Plan de la bibliographie

Tout d'abord, les différentes méthodes d'optimisation existantes seront présentées succinctement, pour ensuite se concentrer sur la réalisation d'une boucle d'optimisation complète. Ensuite, la seconde partie se concentrera sur l'optimisation appliquée à la mécanique des fluides, en commençant par les équations de Navier-Stokes, puis en étendant le sujet aux équations de Lattice-Boltzmann qui sont celles utilisées pour l'aérodynamique externe ici. Enfin, la dernière partie mettra l'accent sur le logiciel ProLB utilisé chez Renault, ainsi que sur les contraintes de simulation liées à l'étude d'un problème industriel automobile à grandes échelles. Ces différentes parties permettront finalement de justifier les choix effectués lors de la thèse, afin de répondre aux attentes industrielles et académiques tout en respectant les contraintes imposées.

Chapitre 1

Choix d'une méthode d'optimisation

1.1 Méthodes d'optimisation

Un problème d'optimisation se caractérise par :

- Une fonction objectif notée I_0 (ex : traînée, portance)
- Des paramètres d'optimisation notés α (continus, discrets, qualitatifs)
- Éventuellement : des contraintes notées R

Bien qu'il existe un très grand nombre de méthodes d'optimisation qu'il est difficile de classer à cause de leur diversité, nous pouvons néanmoins citer deux principales catégories d'optimisation : les méthodes non déterministes et les méthodes déterministes.

Dans un problème d'optimisation non déterministe, l'algorithme génère des solutions qui ne sont pas nécessairement meilleures que la solution de départ. Les algorithmes évolutionnaires sont un exemple de méthode non déterministe, ils reproduisent l'évolution naturelle (telle que la génétique et les mutations), et les appliquent à un ensemble de solutions candidates, jusqu'à avoir atteint le critère recherché. Puisque le processus d'évolution n'interfère pas avec le solveur, cette méthode n'est pas restrictive à une physique spécifique. Cependant, un nombre élevé de résultats doit être obtenu avant d'évaluer la solution optimale, ce qui peut mener à des temps de calcul très lourds, notamment en mécanique des fluides.

A l'inverse, les algorithmes d'optimisation déterministes améliorent une géométrie initiale en fonction des sensibilités de la fonction objectif par rapport aux paramètres d'optimisation. Puisque ces paramètres ne sont pas mis à jour de façon aléatoire mais sont basés sur la direction donnée par les sensibilités, chaque étape de l'optimisation améliore la géométrie donnée. L'efficacité des méthodes déterministes repose principalement sur la façon de calculer les sensibilités, ou gradients, de la fonction objectif. Un panel des différentes solutions proposées pour calculer ces gradients sera présenté dans

la sous-section 1.2.2.

Quelle que soit la catégorie d'optimisation utilisée, il existe plusieurs façons de modifier la géométrie en question :

- l'optimisation paramétrique repose sur une optimisation des paramètres CAO ou des paramètres de maillage,
- l'optimisation surfacique repose sur une optimisation des frontières de la forme,
- l'optimisation topologique repose sur une optimisation de la topologie et non seulement des frontières (des trous peuvent être créés dans la géométrie par exemple).

1.1.1 Optimisation Paramétrique

L'optimisation de forme paramétrique consiste à paramétrer une forme, et à rechercher une solution optimale pour un objectif donné, dans le sous-espace correspondant à la représentation géométrique choisie. Les formes sont ainsi décrites par un nombre réduit de variables (une épaisseur, un rayon, une hauteur, etc). Ce paramétrage peut être fait dans un modèleur CAO (ex : Catia), ou sur un maillage (on parle alors de morphing).

Très souvent, on utilise le paramétrage d'une forme pour créer un plan d'expériences, c'est-à-dire une suite de calculs créée pour interroger l'espace de design pour cet ensemble donné de paramètres. Le but des plans d'expériences est d'obtenir un maximum d'informations en ne réalisant qu'un minimum de calculs. Cela mène à l'exploitation d'une surface de réponse pour le choix de la meilleure configuration de paramètres CAO permettant de minimiser la fonction objectif choisie. La figure 1.1 représente le processus typique d'une optimisation paramétrique :

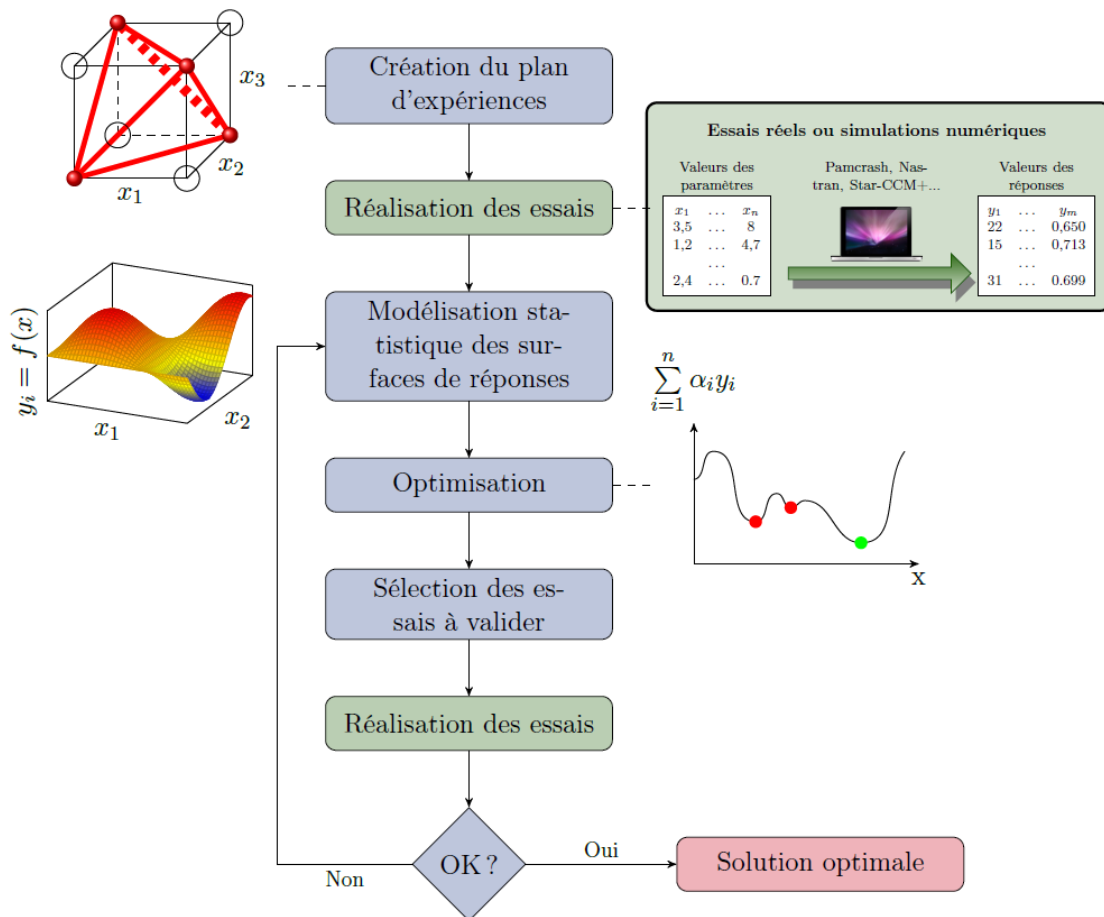


FIGURE 1.1 – Déroulement d’une optimisation par plan d’expériences

Cette démarche est par exemple effectuée dans le logiciel Alternova qui est utilisé chez Renault pour effectuer des optimisations paramétriques sur différentes pièces du véhicule. Cela représente une optimisation non déterministe, puisque l’on génère des géométries qui n’améliorent pas nécessairement le critère choisi et que les valeurs des sensibilités de chaque paramètre d’optimisation ne sont pas connues. Il faut noter que l’espace de recherche de la solution optimale est limité par le nombre de paramètres, le coût de calcul de l’optimisation y étant directement lié et proportionnel.

Paramétrage sur la CAO

Le paramétrage de la CAO est généralement fait sur des pièces spécifiques de la voiture, telles que le rétroviseur ou la jante, qui a déjà fait l’objet de précédentes thèses chez Renault ([24]). L’inconvénient de cette méthode est que le paramétrage peut s’avérer difficile si l’on travaille avec des géométries complexes. Cette approche est donc bien

adaptée à des géométries simples et facilement paramétrables, dont le nombre de paramètres est raisonnable. Ce paramétrage est fait sur le logiciel CATIA chez Renault

Paramétrage sur le maillage

Un autre ensemble de paramètres d'optimisation concernent les paramètres d'un maillage et non d'une CAO. De la même façon, le paramétrage du maillage peut être utilisé pour créer un plan d'expériences afin d'optimiser le critère recherché. Le fait de paramétrer le maillage est intéressant car en déplaçant des nœuds de contrôle bien choisis (les paramètres en question de l'optimisation), cela permet de déplacer le maillage entier. Deux techniques de paramétrage sur le maillage seront présentées ici.

Interpolation multiquadratique dans Star-CCM+

Dans STAR-CCM+, un ensemble de points de contrôle définit le mouvement que le morpher impose au maillage. Chaque point de contrôle est associé à un vecteur déplacement que le morpher utilise pour déplacer les nœuds voisins.

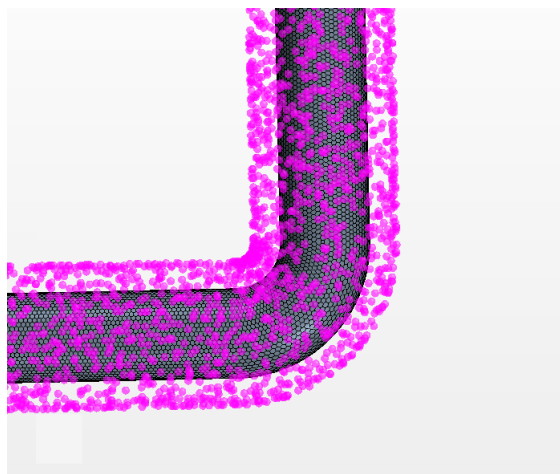


FIGURE 1.2 – Morphing dans Star-CCM+

En utilisant la théorie de l'interpolation multiquadratique [33], ces déplacements sont ensuite utilisés pour générer un champ d'interpolation qui s'applique au maillage entier. Une fois que ce champ a été généré, le morpher l'utilise pour déplacer les nœuds du maillage à leur nouvelle position. Pour générer le champ d'interpolation, un système d'équations est créé en utilisant les nœuds de contrôle x_i et leur déplacement qui est connu d'_i :

$$\mathbf{d}'_i = \sum_{j=1}^N f_{b,j}(r_{ij}) \boldsymbol{\lambda}_j + \boldsymbol{\alpha} \quad (1.1)$$

avec $f_{b,j}(r_{ij})$ une fonction de base radiale de la forme :

$$f_{b,j}(r_{ij}) = \sqrt{(r_{ij}^2 + c_j^2)} \quad (1.2)$$

et r_{ij} la distance entre deux points :

$$r_{ij} = |\mathbf{x}_i - \mathbf{x}_j| \quad (1.3)$$

Dans ces équations, λ est le coefficient d'expansion, x_i est la position du nœud i , N est le nombre de points de contrôle, α est un vecteur constant et c_j est une constante fixée à 0 dans Star-CCM+.

La constante α est utilisée pour satisfaire la contrainte :

$$\sum_{j=1}^N \lambda_j = 0 \quad (1.4)$$

Les équations 1.1 et 1.4 sont ensuite résolues pour trouver λ_j et α . Il est alors possible de calculer les déplacements des nœuds du maillage.

$$\mathbf{d}'(\mathbf{x}) = \sum_{j=1}^N f_{b,j}(r) \lambda_j + \alpha \quad (1.5)$$

avec $f_{b,j}(r)$ qui s'applique à tous les nœuds du maillage :

$$f_{b,j}(r) = \sqrt{(r^2 + c_j^2)} \quad (1.6)$$

et r_{ij} la distance entre deux nœuds :

$$r = |\mathbf{x} - \mathbf{x}_j| \quad (1.7)$$

Polynômes de Bernstein dans Fluent

Dans le logiciel Fluent [1], le morphing est fait de la façon suivante : un volume de contrôle rectangulaire englobe la surface, ou une partie de la surface, de l'objet dont on veut modifier la forme. Un ensemble régulier de points de contrôle est ensuite distribué dans ce volume de contrôle.

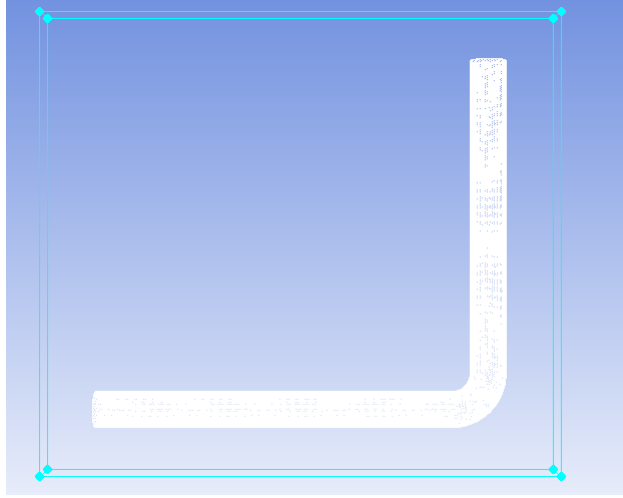


FIGURE 1.3 – Morphing dans Fluent

Les propriétés des polynômes de Bernstein sont utilisées pour définir un repère local. Les coordonnées standards, \mathbf{x}_k , de chaque nœud interne et surfacique du maillage, à l'intérieur du volume de contrôle, sont définies par des coordonnées locales : (u_k, v_k, w_k) . Dans le volume de contrôle, avec $l \times m \times n$ points de contrôle, la relation entre la position du k -ième nœud du maillage et la position du ijk -ième nœud de contrôle est la suivante :

$$\mathbf{x}_k = \sum_{i,j,k=0}^{l,m,n} \xi_{ijk} B_{i,l}(u_k) B_{j,m}(v_k) B_{k,n}(w_k) \quad (1.8)$$

avec ξ_{ijk} les coordonnées du ijk -ième point de contrôle et $B_{i,l}(u)$ le i -ième polynôme de Bernstein de degré l :

$$B_{i,l}(u) = \binom{l}{i} u^i (1-u)^{l-i} \quad (1.9)$$

Si l'ensemble régulier de points de contrôle est modifié en déplaçant un ou plusieurs points de contrôle, la relation précédente permet de repositionner sans irrégularité les nœuds du maillage.

1.1.2 Optimisation Surfaccique par morphing direct

L'optimisation surfaccique consiste, à partir d'une forme initiale, à faire varier la position des frontières de la forme (on déplace des nœuds de la frontière). Cela est fait via une optimisation déterministe, au cours de laquelle on calcule les sensibilités sur chaque nœud du maillage de la forme. On sait de cette façon où déformer l'objet pour optimiser les performances aérodynamiques par exemple. Il faut alors savoir comment réaliser cette déformation. Pour cela, on peut modifier la géométrie via un morpher qui déplace chaque nœud du maillage surfaccique.

Morphing Direct

Le morphing direct, contrairement aux boîtes de morphing ou à l'interpolation quadratique expliquées dans la partie précédente, permet de déplacer tous les nœuds du maillage surfacique individuellement. Le principal avantage est que le morphing direct ne nécessite pas la création de boîtes de morphing, et il est donc plus facile à mettre en place. Cependant le paramétrage du maillage est intéressant car il possède la caractéristique intrinsèque de lisser les sensibilités (qui sont très souvent bruitées dans les cas industriels), ce qui n'est pas le cas pour le morphing direct. Pour cette raison, il est nécessaire d'utiliser une étape de lissage des sensibilités avant d'utiliser le morphing direct.

Etape de lissage des sensibilités

Il existe de nombreuses méthodes pour lisser des sensibilités, seules celles utilisées dans le cadre de la thèse seront expliquées ici.

Neighbor-based smoothing

Le lissage en utilisant les points voisins permet d'évaluer une valeur comme la moyenne pondérée des valeurs voisines [56]. Les poids sont définis tels que les points les plus proches ont les plus gros poids. La valeur estimée est donc lissée, via deux paramètres : le nombre de points voisins choisis, et le nombre d'itérations du lissage.

Feature-based smoothing

Ce type de lissage permet de garder les principales caractéristiques d'une forme, telles que la courbure [95]. La modélisation est basée sur les caractéristiques d'une surface, qui sont préservées ou contrôlées explicitement. Pour cela, une définition mathématique basée sur les lignes pour lesquelles la courbure principale est maximale ou minimale est utilisée (crêtes et creux). Une telle description permet de conserver des informations de qualité concernant la forme étudiée. Les caractéristiques d'une forme, telle que courbure minimale ou maximale, sont donc conservées dans la nouvelle forme.

L'optimisation surfacique permet ainsi de modifier les frontières de la forme d'un objet comme nous l'avons vu. La dernière partie concerne l'optimisation topologique, qui permet de modifier la topologie d'un objet et pas seulement les frontières de sa forme.

1.1.3 Optimisation Topologique

L'optimisation topologique est une autre technologie d'optimisation de forme, dont le maillage volumique ne change pas comparativement à l'optimisation surfacique. On

économise donc du temps de remaillage, puisque l'optimisation topologique modifie la topologie de l'écoulement sans modifier le maillage volumique. Pour cela, un algorithme, basé sur des gradients ou pas, sédimente des mailles dans le volume total disponible maillé afin de trouver la configuration de matière optimale pour l'objectif recherché.

L'optimisation topologique en structure a été initialement présentée par Bendsoe et Kikuchi [7] en 1988. Au lieu de rechercher la forme optimale d'une structure en faisant évoluer ses frontières, ils ont recherché la meilleure distribution de matière de façon à identifier les endroits où rajouter de la matière pour augmenter la rigidité sous une certaine charge. Il est aussi possible de coupler une méthode level-set avec des méthodes d'optimisation afin de pouvoir suivre les frontières du domaine au cours de l'optimisation [4, 3], [66].

En CFD, la même idée a été appliquée à des écoulements de Stokes en 2003 par Borrvall et Peterson [11], qui ont rajouté un terme dépendant de la porosité α dans les équations d'état. Cette variable de porosité permet alors de solidifier certaines zones de l'écoulement, de façon à optimiser le critère choisi. Les paramètres d'optimisation sont donc ici les porosités de chaque cellule, il y en a autant que de cellules du maillage. Le principal inconvénient de l'optimisation topologique est le fait que l'interface fluide/solide n'est pas connue avec précision au cours du calcul (frontière diffuse sur plusieurs mailles de calcul), ce qui peut entraîner des imprécisions plus ou moins importantes. Pour résoudre ce problème, plusieurs solutions ont été proposées. Une approche a été développée depuis quelques années ([40], [42], [13]), qui n'est pas basée sur la porosité mais sur une méthode level-set permettant de localiser l'interface solide/fluide. Dans cette approche, on recherche la répartition optimale de porosité qui minimise le critère choisi en recherchant les valeurs nulles d'une équation level-set, qui marque clairement la distinction entre solide et fluide.

Sans localisation de l'interface

Helgason [36] a réalisé une thèse intitulée " Development of adjoint-based optimization methods for ducted flows in vehicles", basée sur le solveur adjoint d'OpenFoam. Il a notamment travaillé sur l'optimisation surfacique d'un refroidisseur de vanne EGR d'un véhicule. Il a obtenu une diminution de 6.5% de la perte de charge de ce conduit, en utilisant les sensibilités surfaciques et un outil de morphing. Il a ensuite comparé cette méthode à l'approche topologique sur une géométrie industrielle. Un résultat intéressant est que l'approche topologique n'a pas donné de résultats concluants, contrairement à l'optimisation surfacique. Cela est probablement dû à l'absence de localisation de l'interface comme expliqué précédemment.

Brugières a utilisé une fonction d'interpolation pour la porosité déduite de ([11]) :

$$f(\alpha) = \alpha_{min} - (\alpha_{max} - \alpha_{min}) q \frac{\alpha}{q + 1 - \alpha} \quad (1.10)$$

avec q un coefficient de pénalisation fixé à 0,1 ; $\alpha_{min} = 0$ et $\alpha_{max} = \frac{\nu}{K}$. Cette fonction permet d'obtenir des valeurs binaires de α de façon à ce que les valeurs intermédiaires comprises entre 0 et 1 sont supprimées. De cette façon, la frontière entre fluide et solide est clairement définie et l'extraction de la surface optimale est facilitée.

Les gains obtenus en appliquant leur méthode à un catalyseur ne sont pas concluants puisque la perte de charge augmente après optimisation. Cela peut être expliqué par l'absence de lissage entre le domaine fluide et solide. En effet après optimisation, la frontière présente une forme en "marche d'escaliers" ce qui a tendance à augmenter la perte de charge. Une étape supplémentaire de lissage de la géométrie semble alors nécessaire pour obtenir de meilleurs résultats, ce qui est en cours de développement dans leur méthode. Une autre explication possible est que l'interface n'est pas localisé au cours de l'optimisation, et le maillage n'est donc pas raffiné localement contrairement à l'approche développée dans [72] qui sera présentée dans la partie suivante.

Avec localisation de l'interface

Critère sur la porosité des cellules

Papoutsis-Kiachagias et Giannakoglou ont implémenté une méthode pour localiser la paroi au cours de leurs calculs afin de pouvoir utiliser des lois de paroi conjuguées à des modèles de turbulence précis [72]. En effet, il est primordial de connaître l'emplacement de la paroi afin de pouvoir y appliquer des raffinements locaux et des lois de paroi adaptées. La localisation de l'interface repose sur le critère suivant : soit f la face appartenant à la fois à C_1 et C_2 . Si $\alpha(C_1) \cdot \alpha(C_2) = 0$ et $\alpha(C_1) + \alpha(C_2) > \epsilon$, alors f appartient à l'interface.

Ils peuvent alors calculer la viscosité turbulente "artificielle" à l'interface. Ils ont aussi implémenté un critère de raffinement du maillage à l'interface pour une meilleure précision, selon le principe suivant : toutes les faces appartenant à des cellules à moins de 5 mm de l'interface sont bisectées pendant chaque cycle d'optimisation. Ils ont appliqué leur méthode à plusieurs cas, dont l'optimisation d'un conduit de ventilation passager dans des conditions turbulentes. La perte de charge totale du conduit est réduite de 45% grâce à leur méthode.

Utilisation d'une level-set

Karpouzias et De Villiers [40] ont développé une méthode de type level-set pour modéliser l'interface solide/fluide. Cette méthode permet d'obtenir une interface précise et de

mieux contrôler l'optimisation. La méthode level-set couplée avec une méthode adjointe continue prend pour paramètres d'optimisation les distances entre les centres des cellules et l'interface solide/fluide. Cette interface est obtenue en résolvant une équation de transport de l'interface, ayant pour inconnue la variable level-set ϕ . Le principal avantage de cette méthode repose sur l'obtention d'une distinction nette entre le fluide et le domaine solidifié. Si l'on étend la méthode au développement d'une frontière immergée précise, les distinctions entre optimisation surfacique et topologique s'amincissent, combinant les avantages des deux méthodes : l'inutilité de remailler pour l'optimisation topologique, et l'obtention d'une interface lisse pour l'optimisation surfacique. Cependant, le maillage n'est pas raffiné à l'interface dans leur méthode ce qui peut engendrer des inexactitudes concernant le traitement proche paroi de la turbulence. Un raffinement de maillage local au niveau de l'interface solide/liquide serait donc souhaitable, et constitue potentiellement une piste d'amélioration concernant cette méthode.

En conclusion, le couplage d'une méthode level-set avec l'optimisation topologique ([40]) permet de localiser l'interface solide/fluide très précisément, mais nécessite un raffinement de maillage local. Celui-ci a déjà été implémenté par [16, 2, 17]. Quant à la méthode de localisation de l'interface basée sur la porosité ([72]), elle bénéficie aussi d'un raffinement local qui permet d'utiliser des lois de paroi adaptées.

A présent que les différentes méthodes d'optimisation ont été présentées, la réalisation d'une boucle d'optimisation complète sera présentée dans la sous-partie suivante.

1.2 Boucle d'optimisation complète

Cette sous-partie explique les différentes étapes nécessaires à la réalisation d'une boucle d'optimisation complète en suivant les différentes étapes du schéma 1.4 :

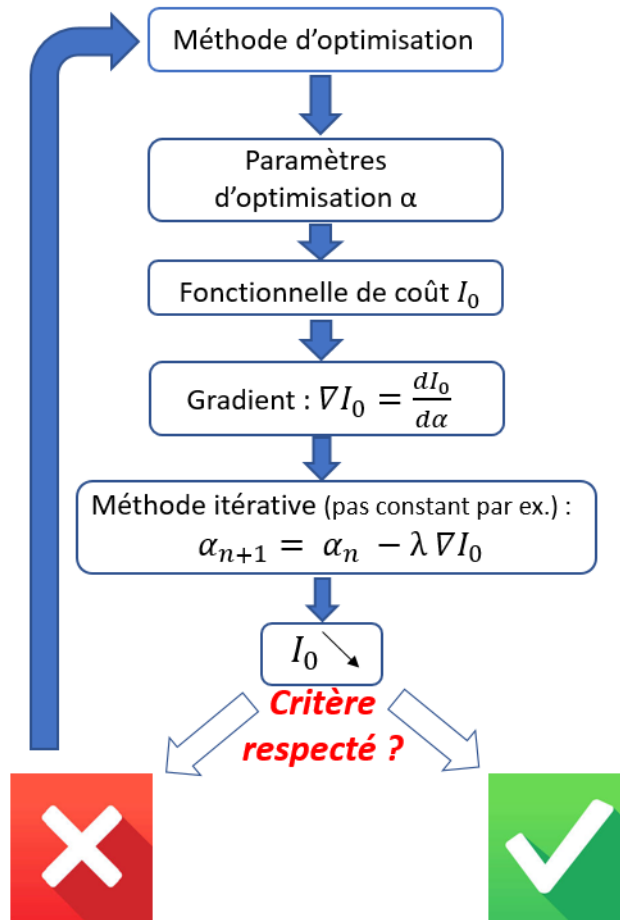


FIGURE 1.4 – Déroulement d'une optimisation numérique

La première étape est de choisir la méthode d'optimisation adaptée à son problème (chapitre 1), puis il faut choisir la fonctionnelle de coût qui nous intéresse (sous-section 1.2.1). Il existe ensuite plusieurs façons d'estimer le gradient de la fonction de coût par rapport aux paramètres d'optimisation (sous-section 1.2.2). Enfin la dernière étape consiste à choisir la méthode itérative permettant de mettre à jour les paramètres d'optimisation pour optimiser la fonction coût (sous-section 1.2.3). Ces différentes sections ne dressent pas une liste exhaustive mais permettent de montrer la démarche à effectuer pour réaliser une boucle d'optimisation complète.

1.2.1 Fonctionnelle de coût I_0

Aérodynamique interne

Dans le cadre de cette thèse, il a été choisi d'étudier des méthodes d'optimisation appliquées à l'aérodynamique interne dans un premier temps. Cela concerne les différents

conduits que l'on retrouve dans une voiture : conduits de climatisation ou conduits moteur par exemples. Les différentes fonctionnelles de coût que l'on peut retrouver dans des conduits seront présentées ici.

Equilibre du débit massique entre plusieurs branches

Un problème qui peut être rencontré en ingénierie automobile est l'obtention un ratio égal de débit massique entre deux sorties de ventilateur par exemple. Le débit massique est défini comme $m = \rho VS$ avec ρ la densité du fluide considéré, V sa vitesse normale moyenne au niveau de la sortie et S la surface de la sortie. Si m_1 correspond au débit massique de la première sortie et m_2 à celui de la deuxième sortie, la fonction objectif à optimiser dans ce cas peut s'écrire de la façon suivante : $I_0 = |m_1 - m_2|$.

Uniformité de l'écoulement

L'uniformité de l'écoulement sortant d'un conduit par exemple est importante pour l'efficacité de la distribution d'air dans la voiture et donc pour le confort du passager. Othmer [69] utilisent l'index d'uniformité γ pour la caractériser grâce à la définition suivante :

$$\gamma = 1 - \frac{1}{2A\bar{u}} \int_{outlet} |u - \bar{u}| dA \quad (1.11)$$

avec \bar{u} la vitesse moyenne normale dans le plan de sortie, et A l'aire de la section de sortie. Une valeur de 1 pour γ correspond à un écoulement parfaitement uniforme.

Contrôle du swirl

Le swirl correspond au moment du débit massique par rapport à un axe défini par un point \mathbf{r}_c et une direction \mathbf{d} . Le swirl est une mesure importante pour l'efficacité des espaces d'entrée d'air, ou orifice d'admission dans les problématiques du remplissage des cylindres des moteurs à explosion. Un swirl élevé est souhaitable pour le mélange efficace de l'air et du carburant dans la chambre de combustion. La formule suivante permet de définir le swirl :

$$\int_V \rho(\mathbf{r} \times \mathbf{u}) \cdot \mathbf{d} dV \quad (1.12)$$

Avec V le volume dans lequel on intègre, et \mathbf{r} la position relative au point \mathbf{r}_c .

Echanges thermiques

Yaji et al. [94] ont formulé un problème d'optimisation visant à maximiser les échanges thermiques pour un écoulement en conduit, pouvant être appliqué à des systèmes de refroidissement. Ils introduisent le coefficient de génération de chaleur β_γ , qui dépend du

paramètre d'optimisation γ de la façon suivante :

$$\beta_\gamma = \beta_{max} + (\beta_{min} - \beta_{max}) \frac{\gamma(1+q)}{\gamma+q} \quad (1.13)$$

Puissance dissipée

La puissance dissipée est la fonction que l'on cherche à minimiser dans les conduits étudiés, à savoir la puissance dissipée due à la perte de pression entre l'entrée et la sortie. Cette fonction s'écrit de la façon suivante :

$$I_0 = - \int_{\Gamma} (p + \frac{1}{2}v^2) \mathbf{v} \cdot \mathbf{n} d\Gamma \quad (1.14)$$

Aérodynamique externe

Sur les applications étudiées en mécanique des fluides chez Renault ; outre les applications en aérodynamique interne, une autre fonctionnelle de coût intéressante est la force de traînée sur la voiture.

Les efforts aérodynamiques sur un véhicule peuvent être calculées de deux façons selon Hucho [38] :

- du point de vue du véhicule,
- du point de vue de l'écoulement

En effet, selon la troisième loi de Newton, les efforts sur le corps et sur l'écoulement sont égaux et opposés.

Intégration sur l'obstacle

Du point de vue du véhicule, la force aérodynamique est l'intégrale sur la surface mouillée des efforts de pression et de viscosité appliqués par le fluide sur le corps. Si on calcule la composante horizontale de cette force on obtient la traînée, si on calcule sa composante verticale on obtient la portance. Cette approche est utilisée dans la plupart des codes commerciaux de mécanique des fluides (Fluent, Star-CCM+, ProLB), et nécessite de faire des interpolations des valeurs de pression et de gradient de vitesse sur chaque maille surfacique du véhicule. En effet, la force s'écrit dans ce cas de la façon suivante :

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \iint_{S_0} p \mathbf{n} dS_0 - \iint_{S_0} \tau \cdot \mathbf{n} dS_0 \quad (1.15)$$

avec p la pression statique, τ le tenseur des contraintes visqueuses et \mathbf{n} la normale unitaire à la maille rentrant dans le solide. Le tenseur des efforts visqueux s'écrit : $\tau = \nu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ et S_0 la surface de l'obstacle.

Cette méthode de calcul des forces aérodynamiques présente l'inconvénient de dépendre fortement du maillage volumique proche de la paroi, ainsi que des schémas d'interpolation utilisés pour déterminer les valeurs de pression et vitesse sur les faces de l'objet.

Intégration sur un volume de contrôle

Les forces aérodynamiques agissant sur un obstacle peuvent être calculées à partir de l'équation de conservation de quantité de mouvement, appliquée à un grand volume de contrôle englobant l'obstacle [8, 18, 86]. Cette formulation est souvent utilisée dans l'industrie automobile pour estimer la traînée d'un véhicule [38], [81] :

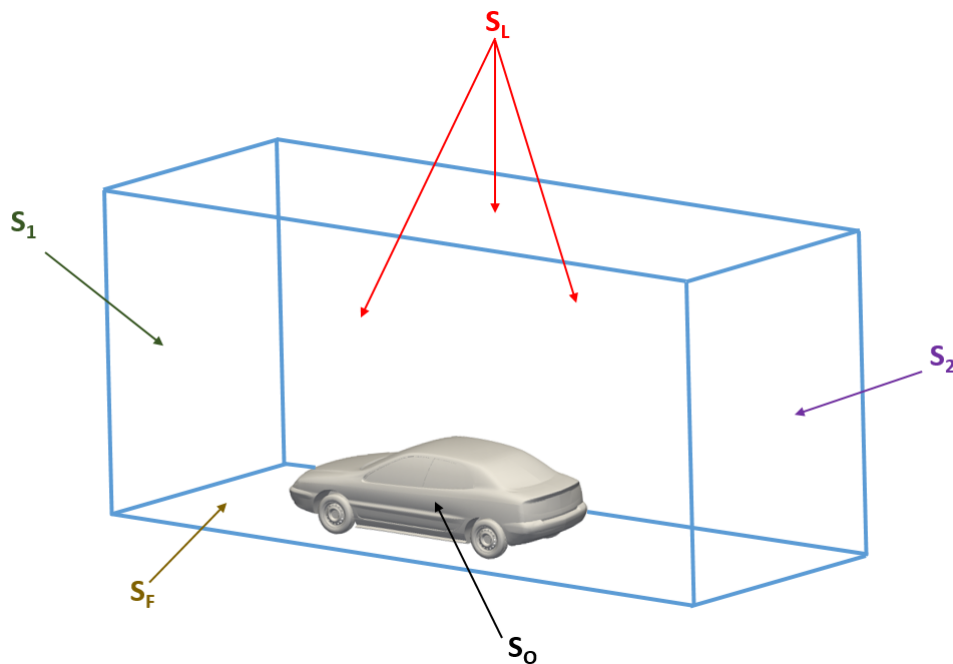


FIGURE 1.5 – Volume de contrôle autour du véhicule

La surface S peut être décomposée en : $S = S_1 + S_2 + S_L + S_O + S_F$ avec S_1 la surface d'entrée, S_2 la surface de sortie, S_L la surface latérale, S_O la surface de l'obstacle et S_F la surface du sol. En orientant le repère direct $O(\mathbf{x}, \mathbf{y}, \mathbf{z})$ tel que $\mathbf{u}_0 = u_0 \mathbf{x}$, nous avons : $\mathbf{n} = -\mathbf{x}$ sur S_1 , $\mathbf{n} = \mathbf{x}$ sur S_2 , $\mathbf{n} = -\mathbf{z}$ sur S_F , et $\mathbf{n} \cdot \mathbf{x} = 0$ sur S_L .

L'équation de conservation de quantité de mouvement peut ensuite être simplifiée grâce à différentes hypothèses que nous énumérons ici.

- on suppose que l'écoulement est incompressible : $\nabla \cdot \mathbf{u} = 0$,
- les effets de gravité sont considérés comme négligeables,
- nous nous intéressons aux efforts aérodynamiques moyens, donc on prend la moyenne temporelle de l'équation de conservation de quantité de mouvement.

Cela donne $\frac{\partial \mathbf{u}}{\partial t} = 0$. On néglige par ailleurs les termes turbulents liés au tenseur des efforts de Reynolds. Nous considérons à présent des valeurs moyennes dans les équations, mais nous gardons les notations telles quelles par souci de simplicité d'écriture.

Ces différentes hypothèses permettent d'écrire l'équation de conservation de mouvement de la façon suivante :

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \tau \quad (1.16)$$

avec τ le tenseur des contraintes visqueuses. En utilisant le théorème de Green-Ostrogradski, on peut écrire :

$$\iiint_V (\mathbf{u} \cdot \nabla) \mathbf{u} dV = \iint_S \mathbf{u}(\mathbf{u} \cdot \mathbf{n}) dS \quad (1.17)$$

avec $\nabla \cdot \mathbf{u} = 0$ (fluide incompressible) et V le volume englobant l'obstacle, S sa surface. Le théorème de Green-Ostogradski donne aussi :

$$\iiint_V \nabla p dV = \iint_S p \mathbf{n} dS \quad (1.18)$$

$$\iiint_V \nabla \cdot (\tau) dV = \iint_S \tau \cdot \mathbf{n} dS \quad (1.19)$$

En reprenant la décomposition de la surface $S : S = S_1 + S_2 + S_L + S_O + S_F$ et en considérant les trois hypothèses suivantes :

- la surface d'entrée S_1 est placée suffisamment loin de l'obstacle en amont pour considérer que $\mathbf{u} = \mathbf{u}_0$ et $p = p_0$ sur S_1 ,
- $\mathbf{u} \cdot \mathbf{n} = 0$ sur S_L (vitesse orthogonale à la normale de la surface),
- la vitesse est égale à 0 sur le sol et sur l'obstacle (condition d'adhérence),
- la surface d'entrée est égale à la surface de sortie : $S_1 = S_2$

On obtient la relation suivante :

$$\iint_S \mathbf{u}(\mathbf{u} \cdot \mathbf{n}) dS = -S_1 u_0^2 \mathbf{x} + \iint_{S_2} \mathbf{u} u_x dS_2 \quad (1.20)$$

Si l'on suppose que $S_1 = S_2$, on obtient :

$$\iint_S \mathbf{u}(\mathbf{u} \cdot \mathbf{n}) dS = \iint_{S_2} u_x (\mathbf{u} - \mathbf{u}_0) dS_2 \quad (1.21)$$

La même décomposition permet d'écrire :

$$\iint_S p \mathbf{n} dS = \iint_{S_2} (p - p_0) \mathbf{x} dS_2 + \iint_{S_L} p \mathbf{n} dS_L + \iint_{S_O} p \mathbf{n} dS_O - \iint_{S_F} p z dS_F \quad (1.22)$$

Concernant le tenseur des contraintes visqueuses, on suppose que la vitesse est uniforme sur la surface d'entrée et qu'il n'y pas de couche limite sur le sol, soit $\iint_{S_1} \tau \cdot \mathbf{n} dS_1 = 0$. On émet la même hypothèse sur les surfaces latérales du volume de contrôle : $\iint_{S_L} \tau \cdot \vec{\mathbf{n}} dS_L = 0$. Sur le sol, le terme visqueux est du type $\frac{\partial u_x}{\partial z}$ et il est associé au

gradient de vitesse dû à la couche limite sur le sol. Dans le cas d'un sol "défilant", on peut écrire : $\iint_{S_F} \tau \cdot \mathbf{n} dS_F = 0$. Enfin, le terme visqueux sur la surface de sortie peut être négligé si l'on se place suffisamment loin dans le sillage, soit $\iint_{S_2} \tau \cdot \vec{n} dS_2 = 0$.

En reprenant la définition de la force aérodynamique totale énoncée dans la partie précédente, on a :

$$\begin{aligned} \mathbf{F} &= \iint_{S_O} p \mathbf{n} dS_O - \iint_{S_O} \tau \cdot \mathbf{n} dS_O \\ &= -\rho \iint_{S_2} u_x (\mathbf{u} - \mathbf{u}_0) dS_2 - \iint_{S_2} (p - p_0) \mathbf{x} dS_2 - \iint_{S_L} p \mathbf{n} dS_L + \iint_{S_F} p \mathbf{z} dS_F \end{aligned} \quad (1.23)$$

En ne gardant que la composante x de ce vecteur pour obtenir la force de traînée :

$$F_x = \rho \iint_{S_2} (u_x u_0 - u_x^2) dS_2 + \iint_{S_2} (p_0 - p) dS_2 \quad (1.24)$$

En introduisant la pression totale $p_t = p + \frac{1}{2} \rho u^2$ et en développant le carré de la vitesse : $u^2 = u_x^2 + u_y^2 + u_z^2$, on obtient l'expression d'Onorato [65] pour définir la traînée, qui dépend seulement de la vitesse et de la pression dans le sillage en aval du véhicule :

$$F_x = -\frac{\rho u_0^2}{2} \iint_{S_2} \left(1 - \frac{u_x}{u_0}\right)^2 dS_2 + \frac{\rho u_0^2}{2} \iint_{S_2} \left(\frac{u_y^2}{u_0^2} + \frac{u_z^2}{u_0^2}\right) dS_2 + \iint_{S_2} (p_{t0} - p_t) dS_2 \quad (1.25)$$

avec pour rappel p_{t0} la pression totale de référence, u_0 la vitesse d'écoulement de référence, ρ la densité de l'air, p_t la pression totale et u_x , u_y et u_z les composantes du vecteur vitesse.

Cette expression est intéressante puisqu'elle permet de décomposer la force de traînée en ses composantes physiques [14], [88], [62], [63]. Le premier terme représente la traînée associée au déficit de vitesse longitudinale mesurée proche du sillage. Elle est associée au développement de tourbillons transversaux proches du sol. Ce terme est très faible en dehors du sillage. Le deuxième terme correspond au développement de tourbillons longitudinaux sur la géométrie et est responsable d'environ 20% de la traînée. Enfin, le dernier terme exprime la perte de pression totale entre l'amont et l'aval du véhicule, due aux phénomènes de sillage tels que la formation et maintenance de structures tourbillonnantes séparées du sillage (décollement, recirculation). Ce dernier terme est le plus important, et peut représenter jusqu'à 80% de la traînée totale.

1.2.2 Estimation du gradient

Une fois que l'on choisit la fonction à optimiser, il est nécessaire de connaître son gradient par rapport aux paramètres d'optimisation. Les différentes façons d'estimer le gradient seront présentées dans la partie suivante, en commençant par la méthode heuristique qui consiste à estimer le gradient sans le calculer réellement.

Méthode heuristique

L'optimisation sans gradient est basée sur la physique de l'écoulement, et ne fait pas appel à des gradients calculés mathématiquement pour savoir comment disposer au mieux du volume alloué. Le logiciel Tosca Fluid propose ce type d'heuristique, dont le principe est le suivant : sachant que les zones de recirculations sont en grande partie responsables des pertes de charge dans un écoulement, leur suppression permettrait d'atteindre de meilleures performances énergétiques. Pour cela, il est nécessaire de connaître leurs emplacements et de trouver un critère permettant de les supprimer. Le critère proposé par [59] est simple : en comparant un écoulement irrotationnel (sans recirculation) à l'écoulement réel, il est possible de localiser les cellules dégradant les prestations attendues et de pénaliser ces cellules (c'est à dire les rendre rigide au sens de l'optimisation topologique). La déviation entre les lignes de courant de l'écoulement irrotationnel et celles de l'écoulement réel permet ainsi directement de pénaliser les cellules souhaitées. Avec un tel raisonnement, il est clair qu'un écoulement sans zone de recirculation ne sera pas optimisable car l'algorithme ne viendra sédimenter aucune cellule.

Une certaine analogie entre le principe du logiciel Tosca Fluid et un algorithme proposé par Othmer ([69]) peut être faite. En effet, Othmer a développé un algorithme d'optimisation topologique sans gradient permettant de supprimer les zones de recirculation d'un écoulement. Pour cela, il détermine un premier écoulement supposé parfait (sans recirculation) régi par l'équation de Laplace pour le potentiel ϕ des vitesses :

$$\Delta\phi = 0 \tag{1.26}$$

Le champ de vitesses "parfait" s'écrit alors :

$$\mathbf{v}_{pot} = -\nabla\phi \tag{1.27}$$

L'écoulement réel est ensuite comparé à l'écoulement dit "parfait" en calculant le produit scalaire entre le vecteur vitesse réel et celui parfait dans chaque cellule. Si la déviation est petite entre ces deux vecteurs, le produit scalaire sera positif et la cellule reste fluide. Si au contraire la déviation est grande, le produit scalaire sera négatif et la cellule sédimentée. L'algorithme pour déterminer le champ de porosité α s'écrit ainsi :

$$\alpha_{n+1} = \alpha_n - \lambda \mathbf{v}_{pot} \cdot \mathbf{v} \tag{1.28}$$

avec λ une constante à déterminer en fonction du cas étudié, les cellules fluides étant celles dont α est petit et les cellules solides celles dont α est grand.

L'inconvénient de cette méthode est qu'elle est difficilement applicable à d'autres critères que la perte de charge, et qu'elle ne fonctionne pas lorsqu'on est en présence d'un conduit avec plusieurs sorties [47]. Cependant certains constructeurs automobiles tels que Ford ont testé cette méthodologie sur des cas simples et ont obtenu des résultats satisfaisants, mais ils n'ont pas précisé s'ils sont arrivés jusqu'à l'industrialisation

de cette méthode sachant que les formes obtenues après optimisation sont difficilement usinables. ([37])

Il semble donc que Tosca Fluid propose une méthodologie intéressante mais qui n'est probablement pas encore assez mûre pour être implémentée dans le processus de fabrication de conduits aérauliques chez Renault. Ceux-ci peuvent en effet s'avérer trop complexes pour être gérés par cette méthode.

Différences finies

On définit la fonction objectif comme $I_0(u(\alpha), \alpha)$. Le gradient s'écrit alors :

$$\frac{dI_0}{d\alpha} = \lim_{h \rightarrow 0} \frac{I_0(\alpha + h) - I_0(\alpha)}{h} \text{ pour chaque } \alpha \quad (1.29)$$

Ce calcul peut s'avérer trop lourd et inefficace pour des cas industriels avec un grand nombre de paramètres d'optimisation (appel au solveur n+1 fois). En effet, il faudrait recalculer l'écoulement à chaque fois qu'on modifie un paramètre d'optimisation (par exemple un nœud du maillage surfacique). De plus, le choix du pas h nécessite une étude de convergence pour en choisir un adapté à son problème, ce qui constitue un problème d'optimisation en soi.

Gradient incomplet

Le gradient de la fonction coût s'écrit classiquement :

$$\nabla I_0 = \frac{dI_0}{d\alpha} = \frac{\partial I_0}{\partial \alpha} + \frac{\partial I_0}{\partial u} \frac{du}{d\alpha} \quad (1.30)$$

Le gradient incomplet est une méthode approchée pour calculer le gradient d'une fonction coût par rapport à un paramètre d'optimisation, en supposant qu'en déplaçant un nœud du maillage surfacique de l'obstacle d'un pas epsilon très petit, l'écoulement reste inchangé. Concrètement, cela revient à supposer que $\frac{du}{d\alpha} = 0$ dans l'équation 1.30 et donc à supposer que $\nabla I_0 = \frac{dI_0}{d\alpha} = \frac{\partial I_0}{\partial \alpha}$. Cela permet de simplifier grandement les calculs, puisque le gradient repose alors sur des critères géométriques et le deuxième terme de 1.30 n'est ainsi pas calculé. Mohammadi [57] montre dans son article l'efficacité de cette méthode, tout en précisant qu'en certains points de l'obstacle (trailing ou leading edge), le gradient est faux et donc inutilisable. Mohammadi et Pironneau [58] couplent la méthode du gradient incomplet avec des modèles à complexité réduite pour simuler le comportement du fluide, et le domaine d'application de la méthode est ainsi moins restreint. Cependant la méthode du gradient incomplet, même si elle est couplée à un modèle réduit, est moins précise que la méthode adjointe qui prend en compte les équations d'état complètes décrivant le comportement du fluide.

Différentiation complexe

La différentiation complexe a été introduite par Lyness et Moler [53], et a été plus tard reprise par Squire et Trapp [84] pour calculer des sensibilités dans des solveurs Navier-Stokes. Comme les différences finies, la méthode est basée sur une expansion en série de Taylor d'une fonction f autour du point x . Mais pour une fonction à variables complexes, le gradient est facilement accessible :

$$\phi(x + i\epsilon\delta x) = \phi(x) + i\epsilon \frac{\partial\phi}{\partial x}(x)\delta x + O(\epsilon^2) \quad (1.31)$$

En prenant les parties réelle et imaginaire de l'équation ci-dessus, on obtient une approximation de $\phi(x)$ et de $\frac{\partial\phi}{\partial x}(x)$ avec une erreur de $O(\epsilon^2)$:

$$\phi(x) = \text{Re}(\phi(x + i\epsilon\delta x)) + O(\epsilon^2) \quad (1.32)$$

$$\frac{\partial\phi}{\partial x}(x) = \frac{1}{\epsilon\delta x} \text{Im}(\phi(x + i\epsilon\delta x)) + O(\epsilon^2) \quad (1.33)$$

Anderson et al. [5] ont appliqué la méthode de différentiation complexe à la CFD en utilisant les équations de Navier-Stokes. Vergnault et Sagaut [89] ont quant à eux montré dans leur étude que leurs résultats sont indépendents de ϵ pour $10^{-18} < \epsilon < 10^{-2}$, ce qui montre la robustesse de la différentiation complexe par rapport à ϵ en LBM, déjà démontrée dans des simulations Navier-Stokes. Cependant, cette méthode peut s'avérer lourde pour des cas industriels avec un grand nombre de paramètres d'optimisation. La méthode adjointe, présentée dans la partie suivante, représente alors une alternative intéressante.

Méthode adjointe

Le calcul du gradient par un problème adjoint repose sur la théorie mathématique des multiplicateurs de Lagrange [10]. Dans cette méthode, on définit une fonction coût "augmentée" appelée Lagrangien qui inclut les contraintes dans le problème d'optimisation.

On rappelle que le gradient de la fonction coût s'écrit :

$$\nabla I_0 = \frac{dI_0}{d\alpha} = \frac{\partial I_0}{\partial \alpha} + \frac{\partial I_0}{\partial u} \frac{du}{d\alpha} \quad (1.34)$$

Le terme problématique dans l'équation précédente est la dérivée des variables d'état par rapport aux paramètres d'optimisation : $\frac{du}{d\alpha}$ car il est très lourd à calculer. Le calculer par différences finies ne serait pas envisageable pour des cas industriels, car pour chaque nœud du maillage de l'obstacle déplacé, il faudrait recalculer l'écoulement. Dans un cas industriel, cela devient impossible car il y aurait trop de calculs à faire, ce qui rend la méthode adjointe très intéressante. En effet, tout l'intérêt de la méthode réside

dans la définition des variables adjointes qui permet de rendre le calcul des gradients indépendant du nombre de paramètres d'optimisation.

On introduit le Lagrangien qui permet d'inclure les contraintes R dans le problème d'optimisation :

$$L = I_0 + u^* R \quad (1.35)$$

avec u^* les multiplicateurs de Lagrange (que l'on nomme quantités adjointes ici).

Le point de départ du problème adjoint consiste à considérer les équations d'état (Navier-Stokes ou Lattice-Boltzmann pour la mécanique des fluides) comme des contraintes du problème d'optimisation :

$$R(u(\alpha), \alpha) = 0 \quad (1.36)$$

avec u et α respectivement les variables d'état et les paramètres d'optimisation.

Lorsque le calcul a convergé, $R = 0$ en tout point puisque R représente les résidus des équations d'état que l'on résoud. La dérivée totale de R est par conséquent nulle aussi :

$$\nabla R = \frac{dR}{d\alpha} = 0 \quad (1.37)$$

Cela signifie que les dérivées totales de notre fonction coût de départ et du Lagrangien sont alors égales :

$$\nabla L = \nabla I_0 \quad (1.38)$$

L'astuce de la méthode adjointe vient du fait que comme R est nul partout, on peut choisir les variables adjointes u^* librement. On décide alors de calculer u^* tel que :

$$\frac{\partial L}{\partial u} = 0 \quad (1.39)$$

L'équation précédente signifie que le Lagrangien est indépendant des variables d'état. Résoudre $\frac{\partial L}{\partial u} = 0$ revient à résoudre une équation sur u^* appelée l'équation adjointe.

Il existe deux approches connues pour implémenter un adjoint : l'approche discrète et l'approche continue. Si les équations adjointes sont dérivées des équations directes *analytiques* puis discrétisées dans un second temps, l'approche est appelée continue : on dérive puis on discrétise. De cette façon, un ensemble d'équations continues peut être obtenu pour le problème adjoint, qui peuvent être discrétisées et résolues grâce à des algorithmes similaires à ceux utilisés pour le problème direct. Si au contraire les équations adjointes sont dérivées des équations directes *déjà discrétisées* alors elles sont appelées discrètes : on discrétise puis on dérive. Cette seconde approche peut se baser sur des procédés de différentiations automatiques pour dériver les équations directes discrétisées. Ce processus est très efficace mais aussi plus complexe. Ces deux approches ont déjà été comparées dans le passé, mais aucune d'elle ne semble prévaloir sur l'autre [61].

L'important est d'obtenir les variables adjointes u^* en chaque point du maillage, tout comme on obtient les variables directes u (vitesse) et p (pression) en chaque point du maillage en résolvant les équations de Navier-Stokes par exemple. Il faut cependant noter que les variables adjointes n'ont pas de valeurs physiques mais purement mathématiques.

Une fois que les champs directs et adjoint ont été obtenus, le gradient de I_0 est calculé de la façon suivante :

$$\nabla I_0 = \nabla L = \frac{dL}{d\alpha} = \frac{\partial L}{\partial \alpha} = \frac{\partial I_0}{\partial \alpha} + u^* \frac{\partial R}{\partial \alpha} \quad (1.40)$$

L'avantage de l'approche adjointe devient ici évident : on n'a plus besoin de calculer le terme $\frac{du}{d\alpha}$ qui était problématique car trop lourd à calculer, et les autres termes nécessaires pour le calcul du gradient de I_0 sont peu coûteux.

1.2.3 Méthode itérative

La dernière étape nécessaire pour réaliser une boucle d'optimisation complète est de choisir la méthode itérative qui permet de mettre à jour les paramètres d'optimisation. En effet, après avoir choisi la méthode d'optimisation adaptée à notre problème (partie 1), ainsi que la fonctionnelle de coût qui nous intéresse (sous-section 1.2.1), puis calculer son gradient par rapport aux paramètres d'optimisation (sous-section 1.2.2), il faut maintenant modifier ces paramètres selon la direction donnée par le gradient afin d'optimiser la fonctionnelle de coût. Cette dernière étape, appelée méthode itérative, sera présentée ici.

Méthode de descente à pas constant

La méthode de descente à pas constant est probablement la méthode de descente la plus intuitive et basique. La méthode est fondée sur le fait que la fonction J décroît, dans le voisinage d'un point x , le plus fortement dans la direction opposée à celle du gradient de I_0 en x , à savoir dans la direction $-\nabla I_0$.

Par exemple, pour une optimisation topologique pour laquelle les paramètres d'optimisation sont les porosités de chaque cellule : une fois que les sensibilités (∇I_0) ont été calculées, un algorithme de plus grande pente permet de mettre à jour les porosités :

$$\alpha_{n+1} = \alpha_n - \lambda \nabla I_0 \quad (1.41)$$

avec λ le pas de descente.

Le choix du pas de descente λ est un problème d'optimisation en soi, mais pour les tests effectués ici nous choisirons un pas constant à fixer en fonction de la configuration. Cet algorithme constitue une source d'amélioration puisqu'il est relativement simple mais peu efficace. Il peut donc être remplacé par des algorithmes plus complexes mais plus efficaces car ils convergent plus rapidement vers la solution optimale. Ce constat

a déjà été montré par le passé [28], nous allons présenter les algorithmes du gradient conjugué et de Newton succinctement dans la suite cette partie.

Gradient conjugué

L'algorithme du gradient conjugué permet de converger plus rapidement vers la solution optimale. La direction de descente est calculée grâce à une combinaison linéaire du gradient de l'itération en question et de la dernière direction de descente pour chaque itération h de l'optimisation. Cet algorithme utilise ainsi la valeur des gradients pour déterminer une direction de descente adaptée à chaque itération de l'optimisation. Même si l'algorithme du gradient conjugué est très souvent démontré avec des exemples de problèmes linéaires, nous écrivons ici l'algorithme de façon généralisé, c'est-à-dire pour des problèmes linéaires mais aussi non linéaires. Par souci de brièveté, nous écrivons ici seulement l'algorithme du gradient conjugué, sans toutes les étapes permettant d'y arriver. Pour une lecture détaillée de la méthode du gradient conjugué, le lecteur est dirigé vers [82].

1. Calcul du gradient ∇I_0
2. Recherche linéaire selon la direction p_h pour trouver λ
3. Calcul des paramètres d'optimisation $\alpha_{h+1} = \alpha_h + \lambda p_h$
4. Calcul du pas β
5. Calcul de la direction de descente $p_{h+1} = -\nabla I_0 + \beta p_h$

Les coefficients α et β peuvent se calculer de différentes façons, selon que le problème étudié est linéaire ou non linéaire.

Descente de Newton

Vergnault et Sagaut [90] ont utilisé la méthode de Newton pour leur optimisation. Leur algorithme d'optimisation suit les étapes suivantes pour chaque itération h de l'optimisation :

1. Calcul de la direction de descente $p_h = \nabla I_0(\alpha_h)$
2. Recherche linéaire selon la direction p_h pour trouver le pas de descente β
3. Calcul des paramètres d'optimisation $\alpha_{h+1} = \alpha_h + \beta p_h$
4. Vérification de la convergence

Une fois qu'ils ont obtenu le gradient grâce à la méthode adjointe (étape 1), l'étape 2 équivaut à minimiser :

$$f(\beta) = I_0(\alpha_h + \beta p_h) \tag{1.42}$$

Cela revient à résoudre $f'(\beta) = 0$. Ceci est effectué en utilisant une série de Taylor pour f' :

$$f'(\beta) = f'(0) + \beta f''(0) \tag{1.43}$$

Finalemment nous obtenons le pas de descente obtenu avec la méthode de descente de Newton : $\beta = -\frac{f'(0)}{f''(0)}$.

Les étapes nécessaires pour réaliser une boucle d'optimisation complète ont été présentées précédemment. Leur application à un problème de mécanique des fluides par la méthode adjointe, utilisée dans le cadre de la thèse, sera détaillée dans la partie suivante, en concluant sur la méthode de Lattice-Boltzmann utilisée chez Renault pour simuler l'aérodynamique externe.

Chapitre 2

Application à la mécanique des fluides

2.1 Optimisation avec les équations de Navier-Stokes

Les équations de Navier-Stokes permettent de décrire l'évolution des variables macroscopiques d'un fluide telles que la pression et la vitesse. Ces équations ont été trouvées indépendamment par G.G. Stokes, en Angleterre, et M. Navier, en France, au début du 19^{ème} siècle, et sont des extensions des équations d'Euler puisqu'elles prennent en compte l'effet de la viscosité dans l'écoulement. Elles dépendent du temps et traduisent la conservation de la masse, de la quantité de mouvement et de l'énergie du fluide.

L'optimisation de forme en CFD a ensuite été initiée dans les années 1970 par différents auteurs tels que Lions, Jameson or Pironneau. Lions a développé la théorie du contrôle optimal en 1968 [51], qui consiste à définir la fonction coût comme un écart avec la solution optimale souhaitée. En 1974, Pironneau a ensuite utilisé la théorie du contrôle pour optimiser la traînée d'un objet dans un écoulement homogène, stationnaire et visqueux [76]. Concernant Jameson, il a appliqué l'optimisation de forme aux équations d'Euler equations en 1988 [39]. Quelques décennies plus tard, après la croissance de la puissance de calcul et de la simulation numérique en mécanique des fluides, l'optimisation par méthode adjointe a été étudiée par Borvall et Petersson [11], qui ont développé une optimisation topologique dans des conduits 2D pour réduire la perte de charge. Othmer a ensuite étendu la méthode à des écoulements turbulents pour l'aérodynamique externe et a obtenu des résultats concluants sur des concepts cars Volkswagen [68, 70]. Dilgen et al [20] ont aussi obtenu des résultats concluants concernant l'aérodynamique automobile en utilisant un modèle de turbulence adjoint. Pour une étude bibliographique détaillée de la méthode adjointe appliquée aux équations de Navier-Stokes, le lecteur est dirigé vers l'article de Papoutsis-Kiachagias et de Giannakoglou [72].

Au début de cette thèse, l'optimisation par adjoint appliquée aux équations de Navier-Stokes a été étudiée et a abouti à des bons résultats sur des cas simples puis industriels.

Ces études ont été menées sur plusieurs codes commerciaux possédant déjà un solveur adjoint : Star-CCM+, Fluent, OpenFOAM et Tosca Fluid (le logiciel Tosca Fluid ne possède pas de solveur adjoint à proprement parler mais est basé sur une heuristique de l'écoulement cherchant à s'en rapprocher). Les résultats obtenus ont été concluants et ont montré que la méthode adjointe appliquée aux équations de Navier-Stokes est déjà un sujet relativement mature, même si en pratique son champ d'application réellement déployé industriellement chez Renault se limite pour l'instant à l'aérodynamique interne (optimisation des conduits). Pour cette raison, la thèse a été reorientée vers le développement d'un adjoint avec la méthode de Lattice-Boltzmann comme nous le verrons par la suite. Une démonstration rapide de la théorie de l'optimisation par adjoint sur Navier-Stokes, basée sur les travaux de Othmer, sera donc présentée ici. La partie suivante se concentrera sur le développement d'un adjoint sur Lattice-Boltzmann qui est un domaine beaucoup plus récent et donc plus permissif pour la thèse en termes de recherches et découvertes.

2.1.1 Equations directes

Les équations de Navier Stokes, sous l'hypothèse d'un écoulement stationnaire et incompressible, s'écrivent :

$$(\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p - \nabla \cdot (2\nu D(\mathbf{v})) + \alpha \mathbf{v} = 0 \quad (2.1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2.2)$$

avec ν la viscosité cinématique et D le tenseur du taux de déformation $D(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T)$. La première équation représente la conservation de la quantité de mouvement, et la seconde la conservation de la masse. On suppose ici que l'on souhaite réaliser une optimisation topologique, ce qui explique la présence du terme de Darcy $\alpha \mathbf{v}$ qui est le terme de porosité qui permet de pénaliser les cellules et est donc l'élément central d'une optimisation topologique. On fait cette hypothèse car historiquement, l'optimisation topologique est apparue en premier dans les études d'optimisation en mécanique des fluides avec les équations de Navier-Stokes.

2.1.2 Equations adjointes

Le problème d'optimisation peut s'écrire comme :

$$\text{minimiser } I_0 = I_0(\alpha, \mathbf{v}, p) \text{ contraint à } R(\alpha, \mathbf{v}, p) = 0 \quad (2.3)$$

avec \mathbf{v} et p qui représentent respectivement la vitesse et la pression, et α les variables d'optimisation. $R = (R_1, R_2, R_3, R_4)^T$ représente les contraintes, ici les équations directes, à savoir les équations de Navier-Stokes stationnaires et incompressibles présentées dans 2.1.1.

Pour rappel, la méthode des multiplicateurs de Lagrange est bien adaptée à la résolution d'un problème d'optimisation. Cette méthode permet, en passant d'un problème d'optimisation sous contraintes à un problème sans contraintes, de trouver l'optimum d'une fonction dérivable d'une ou plusieurs variables. Le Lagrangien s'écrit :

$$L = I_0 + \int_{\Omega} (\mathbf{u}, q) R d\Omega \quad (2.4)$$

Résoudre le problème d'optimisation équivaut à rechercher des optimum du lagrangien. Les multiplicateurs de Lagrange sont ici la vitesse adjointe u et la pression adjointe q . On peut écrire :

$$\frac{dL}{d\alpha} = \frac{\partial L}{\partial \alpha} + \frac{\partial L}{\partial \mathbf{v}} \frac{d\mathbf{v}}{d\alpha} + \frac{\partial L}{\partial p} \frac{dp}{d\alpha} \quad (2.5)$$

En reprenant les notations de l'article de Othmer [67], on choisit les variables adjointes telles que :

$$\frac{\partial L}{\partial \mathbf{v}} \frac{d\mathbf{v}}{d\alpha} + \frac{\partial L}{\partial p} \frac{dp}{d\alpha} = \frac{\partial L}{\partial \mathbf{v}} \frac{\delta \mathbf{v}}{\delta \alpha} + \frac{\partial L}{\partial p} \frac{\delta p}{\delta \alpha} = 0 \quad (2.6)$$

Ce qui revient à écrire :

$$\frac{\partial L}{\partial \mathbf{v}} \delta \mathbf{v} + \frac{\partial L}{\partial p} \delta p = 0 \quad (2.7)$$

Nous pouvons à présent déterminer les équations adjointes grâce à l'équation suivante :

$$\frac{\partial I_0}{\partial \mathbf{v}} \delta \mathbf{v} + \frac{\partial I_0}{\partial p} \delta p + \int_{\Omega} (\mathbf{u}, q) \frac{\partial R}{\partial \mathbf{v}} \delta \mathbf{v} d\Omega + \int_{\Omega} (\mathbf{u}, q) \frac{\partial R}{\partial p} \delta p d\Omega = 0 \quad (2.8)$$

Sachant que R représente les équations d'état directes, c'est-à-dire les équations de Navier-Stokes stationnaires et incompressibles, nous arrivons à :

$$\frac{\partial (R_1, R_2, R_3)^T}{\partial \mathbf{v}} \delta \mathbf{v} = (\delta \mathbf{v} \cdot \nabla) \mathbf{v} + (\mathbf{v} \cdot \nabla) \delta \mathbf{v} - \nabla \cdot (2\nu D(\delta \mathbf{v})) + \alpha \delta \mathbf{v} \quad (2.9)$$

$$\frac{\partial R_4}{\partial \mathbf{v}} \delta \mathbf{v} = -\nabla \cdot \delta \mathbf{v} \quad (2.10)$$

$$\frac{\partial (R_1, R_2, R_3)^T}{\partial p} \delta p = \nabla \delta p \quad (2.11)$$

$$\frac{\partial R_4}{\partial p} \delta p = 0 \quad (2.12)$$

La variation de la viscosité turbulente ν_t a été ici négligée. Cette hypothèse connue sous le nom d'hypothèse de "turbulence gelée" est une approximation courante, dont l'influence sur la précision des résultats sera analysée dans la suite du rapport.

L'équation devient alors :

$$\frac{\partial I_0}{\partial \mathbf{v}} \delta \mathbf{v} + \frac{\partial I_0}{\partial p} \delta p + \int_{\Omega} \mathbf{u} \cdot ((\delta \mathbf{v} \cdot \nabla) \mathbf{v} + (\mathbf{v} \cdot \nabla) \delta \mathbf{v} - \nabla \cdot (2\nu D(\delta \mathbf{v})) + \alpha \delta \mathbf{v}) d\Omega - \int_{\Omega} q \nabla \cdot \delta \mathbf{v} d\Omega + \int_{\Omega} \mathbf{u} \nabla \delta p d\Omega = 0 \quad (2.13)$$

La fonction coût I_0 peut se décomposer sur l'intérieur Ω du domaine et sur les bords Γ :

$$I_0 = \int_{\Gamma} I_{0\Gamma} d\Gamma + \int_{\Omega} I_{0\Omega} d\Omega \quad (2.14)$$

Après intégration par parties ([67]), l'équation devient avec la décomposition précédente :

$$\begin{aligned} & \int_{\Gamma} d\Gamma (\mathbf{u} \cdot \mathbf{n} + \frac{\partial I_{0\Gamma}}{\partial p}) \delta p + \int_{\Omega} d\Omega (-\nabla \cdot \mathbf{u} + \frac{\partial I_{0\Omega}}{\partial p}) \delta p + \int_{\Gamma} d\Gamma (\mathbf{n}(\mathbf{u} \cdot \mathbf{v}) + \mathbf{u}(\mathbf{v} \cdot \mathbf{n}) + 2\nu \mathbf{n} \cdot D(\mathbf{u}) - q\mathbf{n} + \frac{\partial I_{0\Gamma}}{\partial \mathbf{v}}) \cdot \delta \mathbf{v} \\ & - \int_{\Gamma} d\Gamma (2\nu \mathbf{n} \cdot D(\delta \mathbf{v}) \cdot \mathbf{u}) + \int_{\Omega} d\Omega (-\nabla \mathbf{u} \cdot \mathbf{v} - (\mathbf{v} \cdot \nabla) \mathbf{u} - \nabla \cdot (2\nu D(\mathbf{u})) + \alpha \mathbf{u} + \nabla q + \frac{\partial I_{0\Omega}}{\partial \mathbf{v}}) \cdot \delta \mathbf{v} = 0 \end{aligned} \quad (2.15)$$

Cette équation est respectée si chaque intégrande est nulle. Les intégrales sur le domaine vont donc nous donner les équations d'état adjointes, et les intégrales sur les bords vont nous donner les conditions limites adjointes. Nous arrivons ainsi aux équations d'état adjointes suivantes :

$$-2D(\mathbf{u})\mathbf{v} = -\nabla q + \nabla \cdot (2\nu D(\mathbf{u})) - \alpha \mathbf{u} - \frac{\partial I_{0\Omega}}{\partial \mathbf{v}} \quad (2.16)$$

$$\nabla \cdot \mathbf{u} = \frac{\partial I_{0\Omega}}{\partial p} \quad (2.17)$$

avec $-2D(\mathbf{u})\mathbf{v} = -\nabla \mathbf{u} \cdot \mathbf{v} - (\mathbf{v} \cdot \nabla) \mathbf{u}$.

L'obtention des conditions limites adjointes est similaire à la démarche précédente, le lecteur est dirigé vers [67] pour les détails théoriques. La structure des équations adjointes est très similaire à celle des équations de Navier-Stokes directes, à l'exception du signe moins devant le terme convectif de l'adjoint qui indique que l'information adjointe est convectée en amont de l'écoulement direct plutôt qu'en aval.

De nombreuses fonctions coût sont calculées sur les bords du domaine, telles que la perte de charge ou l'uniformité des vitesses en sortie, ce qui en termes mathématiques revient à $I_{0\Omega} = 0$ dans les équations adjointes 2.16 et 2.17.

2.1.3 Calcul du gradient

Les sensibilités topologiques sont alors égales à :

$$\nabla I_0 = \nabla L = \frac{\partial I_0}{\partial \alpha} + \int_{\Omega} (\mathbf{u}, q) \frac{\partial R}{\partial \alpha} d\Omega \quad (2.18)$$

La porosité est une variable auxiliaire permettant de décrire une transition continue entre fluide et solide, et il n'y a pas de dépendance usuelle de la fonction coût par rapport à cette porosité : $\frac{\partial I_0}{\partial \alpha_i} = 0$. Il reste à calculer :

$$\frac{\partial R}{\partial \alpha_i} = \begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix} \quad (2.19)$$

La sensibilité de chaque cellule s'écrit alors comme le produit scalaire de la vitesse directe et de la vitesse adjointe multiplié par le volume V_i de la cellule en question :

$$\frac{dI_0}{d\alpha_i} = \mathbf{u}_i \cdot \mathbf{v}_i V_i \quad (2.20)$$

Il apparait donc ici que le calcul de sensibilités ne nécessite qu'un appel au solveur direct pour accéder à \mathbf{u}_i , et un appel à l'adjoint pour accéder à \mathbf{v}_i . Le coût de calcul est donc très intéressant par rapport à un calcul par différences finies.

Selon les travaux de Othmer, il est possible de calculer les gradients surfaciques similairement aux gradients topologiques. En effet, il montre que la sensibilité surfacique pour un déplacement β d'un certain point de la surface vers l'extérieur du domaine peut être approximée par :

$$\frac{\partial I_0}{\partial \beta} \propto \mathbf{u}_t^i \mathbf{v}_t^i = \mathbf{u}^i \mathbf{v}^i \propto \frac{\partial I_0}{\partial \alpha} \quad (2.21)$$

avec \mathbf{u}_t^i et \mathbf{v}_t^i respectivement les vitesses adjointe et directe tangentielles dans la cellule i adjacente à la perturbation de la surface. La dernière proportionnalité est due au fait que les vitesses normales sont nulles proche de la paroi.

De cette façon, un mouvement vers l'intérieur du domaine à une certaine position de la surface a le même effet qu'augmenter la porosité dans la cellule adjacente à cette position. Les deux optimisations (topologiques ou surfaciques) reviennent à pénaliser la cellule, ce qui est cohérent.

2.2 Optimisation avec les équations de Lattice-Boltzmann

2.2.1 Présentation générale de la LBM

Dans la plupart des recherches associées à l'optimisation par méthode adjointe pour des problèmes de mécanique des fluides, les équations de Navier-Stokes sont utilisées. Alternativement, la méthode de Lattice-Boltzmann (LBM) est devenue une approche intéressante en CFD. Elle repose sur une description mésoscopique et une modélisation statistique de l'écoulement. Le point de départ de la LBM est la théorie cinétique des gaz, introduite par Ludwig Boltzmann en 1872. Pour des écoulements à bas nombre de Mach incompressible, la LBM peut approximer les solutions des équations de Navier-Stokes. La méthode est donc intéressante parce que simple algorithmiquement parlant, bien formulée pour la parallélisation, et elle s'applique bien à des écoulements complexes : poreux, multiphasique, miscible, immiscible, etc. La LBM est aussi bien adaptée pour l'étude de problème d'optimisation complexe et à grandes échelles. Nous choisissons d'utiliser la LBM pour simuler l'aérodynamique externe car elle est simple à implémenter de façon parallèle et est très efficace sur un grand nombre de processeurs grâce à son maillage cubique généré automatiquement. La LBM est aussi compatible avec des écoulements faiblement compressibles, dont les écoulements autour de véhicules font partie (faible nombre de Mach).

Chez Renault, le logiciel ProLB qui est basé sur la méthode de Lattice-Boltzmann est utilisé en aérodynamique et aéroacoustique externe. Ce logiciel est basé sur un modèle Boltzmann sur Réseau qui est défini par trois grandes caractéristiques [77], [15] :

- Un modèle de vitesses discrètes c_i avec $i = 1..Q$ dans un espace physique de dimension D . Les modèles les plus connus sont les modèles D2Q7, D2Q9, D2Q13, D3Q15, D3Q19, D3Q27, D3Q39,..
- Un modèle de collision. Le modèle le plus simple est le modèle à un seul temps de relaxation (SRT : Single Relaxation Time) qui représente le modèle de collision BGK de l'équation de Boltzmann. Il existe également des modèles à plusieurs temps de relaxation (MRT : Multiple Relaxation Time),
- Un modèle de fonction d'équilibre. Le modèle le plus simple est basé sur le développement limité de la fonction d'équilibre de Maxwell-Boltzmann tronquée à l'ordre 2. L'ordre de troncature de la fonction d'équilibre est lié au nombre de vitesses discrètes.

L'équation de Boltzmann est une description statistique du mouvement de particules à un niveau mésoscopique. Elle décrit l'évolution d'une fonction de probabilité notée f , qui est la probabilité de présence de particules à un temps t et une localisation x , ayant une vitesse c . L'équation de Boltzmann continue est discrétisée dans l'espace des vitesses pour obtenir la DVBE : Discrete Velocity Boltzmann Equation. Ensuite, on utilise une discrétisation en temps et espace pour obtenir l'équation de Boltzmann sur réseau, ou LBM (Lattice Boltzmann Method). Ici, la discrétisation (ou réseau) D3Q19 est utilisé parce qu'elle est bien adaptée à la simulation d'écoulement athermiques et faiblement compressibles, même si la LBM est une méthode intrinsèquement compressible.

En résumé, la méthode de Lattice-Boltzmann consiste à considérer un ensemble f_i de fonctions de distribution des particules qui possèdent chacune une vitesse donnée par un ensemble restrictif de vecteurs c_i . Le modèle D3Q19 représente ainsi un ensemble de 19 vecteurs vitesses comme le montre la figure suivante :

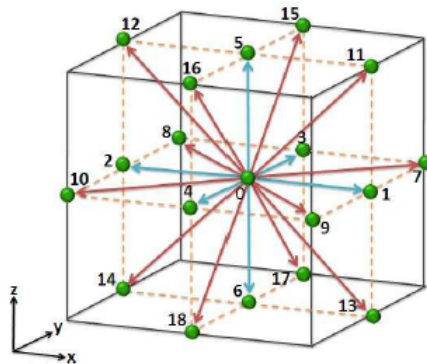


FIGURE 2.1 – Discrétisation des vitesses pour le modèle D3Q19

Les coefficients de pondération pour la quadrature D3Q19 sont $\omega_0 = \frac{1}{3}$ pour le centre, $\omega_{1\dots 6} = \frac{1}{18}$ pour les faces, et $\omega_{7\dots 18} = \frac{1}{36}$ pour les cotés. Le modèle D3Q19 est d'un ordre suffisamment élevé pour retrouver les équations de Navier-Stokes, et permet de calculer les variables macroscopiques (la densité ρ et la vitesse \mathbf{u}) grâce à des sommes discrètes sur l'espace des vitesses :

$$\rho = \sum_i f_i \quad (2.22)$$

$$\mathbf{u} = \frac{1}{\rho} \sum_i f_i \mathbf{c}_i \quad (2.23)$$

Pour une explication détaillée de l'obtention de l'algorithme de Boltzmann sur réseau, le lecteur est dirigé vers le mémoire de thèse de Félix Gendre [27]. Celui-ci contient les différentes étapes permettant d'y arriver (équation continue, équation à vitesses discrètes (DVBE), et enfin équation sur réseau (LBM)).

2.2.2 Etude bibliographique de la méthode adjointe en LBM

Comparé aux études précédentes, l'optimisation de forme basée sur un adjoint de la méthode de Lattice-Boltzmann est un domaine de recherche relativement récent. Une explication possible est le fait que la LBM est utilisée en industrie depuis moins longtemps que des solveurs de type Navier-Stokes. Pour cette raison, les études sur l'optimisation par méthode adjointe en LBM sont aussi plus récentes que celles utilisant les équations de Navier-Stokes.

L'optimisation par adjoint en LBM a été initiée il y a une dizaine d'années par Tekitek, en 2006 [85] (voir tableau ci-dessous pour une analyse de la littérature sur l'optimisation par adjoint en LBM). Il a développé un adjoint discret appliqué à un écoulement de Poiseuille pour identifier des paramètres d'optimisation. En 2008, Pinggen [75] a ensuite obtenu des résultats similaires à ceux obtenus par Borrvall et Peterson [11] avec les équations de Navier-Stokes. Suite à cette étude, Makhija and al. [54] ont introduit une méthode à temps de relaxation multiple appliqué à des dispositifs de mélange en 2012. Concernant Krause et al [43], ils ont développé un adjoint continu, basé sur la forme continue des équations de Boltzmann en 2013.

Cependant, l'approche continue empêche l'utilisation de conditions limites précises puisque les conditions limites de la LBM sont généralement formulées à partir de la forme discrétisée en vitesses de l'équation de Boltzmann (DVBE). Pour contrer cela, Saggaut et al. [90] ont dérivé l'adjoint à partir de la forme discrétisée en vitesses, temps et espace de l'équation de Boltzmann et ont obtenu des équations adjointes dont la forme est similaire à la LBM classique. Ils utilisent donc la LBM et non la DVBE pour formuler le problème adjoint, ce qui est tout aussi valable car les conditions limites peuvent aussi être formulées à partir de la LBM. Cela permet d'obtenir des conditions limites

précises tout en conservant les avantages de l'approche continue, à savoir la possibilité d'utiliser les mêmes schémas numériques pour résoudre la LBM directe et l'adjoint. Mais leur approche peut aussi être considérée comme une approche discrète puisqu'ils utilisent l'équation déjà discrétisée en espace, temps et vitesses pour dériver et trouver leur problème adjoint. Cependant un point intéressant à remarquer est le fait que même si leur approche peut être vue comme une approche discrète, elle n'hérite pas de ses principaux désavantages à savoir la résolution de systèmes linéaires passant par des opérations matricielles complexes. Nous adopterons donc cette approche tout au long de la thèse. Yaji et al [94] ont aussi utilisé cette approche mixte et l'ont étendue à des problèmes d'optimisation thermiques.

Le tableau suivant résume les différents articles traitant d'optimisation par adjoint appliquée à la LBM. Dans ce tableau, dans la colonne CL (Conditions Limites), BB correspond au Bounce-Back, Bouzidi et Guo aux conditions limites tirées de leur étude respective [12] et [31]. La colonne "approximation temporelle" concerne la gestion de la dépendance temporelle des problèmes direct et adjoint (stationnarité ou instationnarité), sur laquelle nous reviendrons dans la section 3.5.

TABLE 2.1 – Récapitulatif de la littérature sur l’adjoint en LBM

auteurs, année	type	formulation adjointe	dimension	fonction coût	turbulence	maillage	CL	collision	approximation temporelle
Tekitek et al. [85], 2005	paramétrique	discret	2D	vitesse cible	non	uniforme	BB	MRT	direct instationnaire-adjoint instationnaire
Pingen et al. [75], 2008	topologique	discret	2D	perte de charge	de non	uniforme	BB	BGK	direct stationnaire-adjoint stationnaire
Kreissl et al. [44], 2009	surfacique	discret	2D	perte de charge	de non	uniforme	Bouzidi	BGK	direct stationnaire-adjoint stationnaire
Makhija et al [54], 2012	topologique	discret	2D	taux de mélange	non	uniforme	BB	MRT	direct stationnaire-adjoint stationnaire
Krause et al. [43], 2013	paramétrique	continue	2D & 3D	vitesse cible	non	uniforme	none	BGK	direct stationnaire-adjoint stationnaire

auteurs, année	type	formulation adjointe	dimension	fonction coût	turbulence	maillage	CL	collision	approximation temporelle
Vergnault et al [90], 2014	paramétrique	mixte	2D	pression quadratique moyenne	non	uniforme	BB	BGK	direct instationnaire-adjoint instationnaire
Liu et al [52], 2014	topologique	discret	2D	dissipation de l'énergie	non	uniforme	BB	MRT	direct stationnaire-adjoint stationnaire
Hekmat et al. [35], 2014	paramétrique	discret	2D	pression et vitesse cible	non	uniforme	BB	BGK	direct instationnaire-adjoint instationnaire
Yaji et al. [94], 2015	topologique	mixte	2D	perte de charge, échanges thermiques	non	uniforme	BB	BGK	direct stationnaire-adjoint stationnaire

auteurs, année	type	approche	dimension	fonction coût	turbulence	maillage	CL	collision	approximation temporelle
Norgaard et al [64], 2015	topologique	discret	2D	vitesse cible	non	uniforme	BB	BGK	direct instationnaire- adjoint ins- tationnaire
Laniewski-Wollk [45], 2015	topologique	discret	2D & 3D	taux de mélange, échanges thermiques	de non	uniforme	BB	MRT	direct stationnaire- adjoint station- naire
Chen et al. [15], 2017	topologique	mixte	2D & 3D	perte de charge	de non	uniforme	BB	BGK	direct quasi- instationnaire- adjoint station- naire
Li et al. [49], 2018	surfactive	mixte	2D	pression cible	non	non uni- forme	Guo	BGK	direct stationnaire- adjoint station- naire
Dugast et al. [21], 2018	topologique	mixte	2D	température moyenne, échanges thermiques	non	uniforme	BB	BGK	direct stationnaire- adjoint station- naire

Sachant que l'objectif final de la thèse est d'optimiser la forme d'un véhicule tout en conservant sa forme globale, il a été choisi de développer une méthode d'optimisation surfacique et non topologique. En effet, l'optimisation surfacique déplace les frontières d'une forme en conservant sa forme globale, alors que l'optimisation topologique modifie la topologie d'une forme en y créant des trous par exemple. Concernant l'optimisation par méthode adjointe en LBM, ce tableau montre que l'optimisation surfacique a été peu étudiée en LBM, à l'exception d'études 2D, avec le modèle de collision BGK, difficilement utilisable pour des cas industriels complexes. A la connaissance de l'auteur, l'optimisation surfacique par méthode adjointe en LBM pour l'étude d'un cas industriel complexe, en régime turbulent et instationnaire, n'a encore jamais été réalisée et constitue un sujet de recherche novateur. La thèse se focalisera donc sur le développement d'un solveur adjoint surfacique en LBM, bien adapté aux problématiques de réduction de la traînée en modifiant la forme du véhicule sans changer sa topologie.

2.2.3 Equations directes

En utilisant l'approximation de BGK pour le terme de collision entre les particules, le comportement des particules est décrit par l'équation de Boltzmann discrétisée en espace, en temps, et en vitesse :

$$f_i(\mathbf{x}_k + \mathbf{c}_i \Delta t, t_{n+1}) = f_i(\mathbf{x}_k, t_n) - \frac{1}{\tau} [f_i(\mathbf{x}_k, t_n) - f_i^{eq}(\mathbf{x}_k, t_n)] \quad (2.24)$$

où f_i est la fonction de distribution, f_i^{eq} est la fonction d'équilibre, τ est le temps de relaxation. La fonction d'équilibre (basé sur le développement limité de la fonction d'équilibre de Maxwell-Boltzmann tronquée à l'ordre 2) s'écrit :

$$f_i^{eq}(\mathbf{x}_k, t_n) = \rho \omega_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right) \quad (2.25)$$

avec u la vitesse du fluide et ρ sa densité.

En utilisant $\mathcal{H}_i^{(2)}$ qui est le polynôme de Hermite du second ordre : $\mathcal{H}_{i\alpha\beta}^{(2)} = c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}$, avec les indices α et β qui correspondent aux directions cartésiennes, on peut aussi écrire la fonction d'équilibre sous la forme :

$$f_i^{eq}(\mathbf{x}_k, t_n) = \rho \omega_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{u_\alpha u_\beta \mathcal{H}_{i\alpha\beta}^{(2)}}{2c_s^4} \right) \quad (2.26)$$

Pour rappel, ces variables macroscopiques décrivant le fluide sont obtenues à partir des fonctions de distributions de la façon suivante :

$$\rho = \sum_{i=1}^{19} f_i \quad (2.27)$$

$$\mathbf{u} = \frac{1}{\rho} \sum_{i=1}^{19} \mathbf{c}_i f_i \quad (2.28)$$

Il est aussi possible de déduire la pression du fluide à partir de la densité grâce à la formule suivante :

$$p = c_s^2 \rho \quad (2.29)$$

2.2.4 Equations adjointes

Un problème d'optimisation basé sur les équations de Lattice-Boltzmann s'écrit de la même façon qu'un problème basé sur Navier-Stokes, à l'exception que les "contraintes" à respecter sont à présent des équations différentes. Le problème s'écrit alors :

$$\text{minimiser } I_0 = I_0(\alpha, f) \text{ contraint à } R(\alpha, f) = 0 \quad (2.30)$$

avec J la fonction coût, f les fonctions de distribution, et α les variables d'optimisation. R représente les contraintes, ici les équations de Lattice-Boltzmann.

Pour obtenir les équations adjointes à partir des équations directes, nous utilisons la même démarche que dans l'article [90]. L'introduction des multiplicateurs de Lagrange (ou variables adjointes) permet d'introduire les équations de Boltzmann comme des contraintes du problème et la nouvelle fonction coût (ou Lagrangien) s'écrit alors :

$$L = I_0 + f^* R \quad (2.31)$$

Il est alors possible de calculer le gradient indépendamment du nombre de paramètres d'optimisation, ce qui constitue le principal intérêt de la méthode adjointe. Les quantités adjointes sont alors choisies, comme il a été montré dans l'équation 1.39, de façon à ce que le Lagrangien soit indépendant des variables directes :

$$\frac{\partial L}{\partial f} = 0 \quad (2.32)$$

L'astuce à utiliser ici est d'utiliser l'approche continue tout en dérivant les équations directes déjà discrétisées en vitesse, en temps et en espace ([90]). Cela revient à remplacer R par $R_j(\mathbf{x}_k, t_n)$ dans l'équation 2.30 de façon à avoir :

$$R_j(\mathbf{x}_k, t_n) = f_j(\mathbf{x}_k + \mathbf{c}_i \Delta t, t_{n+1}) - f_j(\mathbf{x}_k, t_n) + \frac{1}{\tau} [f_j(\mathbf{x}_k, t_n) - f_j^{eq}(\mathbf{x}_k, t_n)] \quad (2.33)$$

Si l'on dérive en se plaçant à un nœud spécifique du maillage $\mathbf{x} = \mathbf{x}_{k0}$ et à un temps donné $t = t_{n0}$, on a alors :

$$\frac{\partial L}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = \frac{\partial I_0}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} + \sum_j \sum_k \sum_n f_j^*(\mathbf{x}_k, t_{n+1}) \frac{\partial R_j(\mathbf{x}_k, t_n)}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = 0 \quad (2.34)$$

Les sommes sur les indices j, k et n viennent du fait que l'on intègre 2.31 sur le temps, le domaine entier et l'espace des vitesses discrètes. Tout l'intérêt d'utiliser $R_j(\mathbf{x}_k, t_n)$ apparaît alors par la suite. On voit ainsi qu'il ne reste quasiment que des Dirac lors de la

dérivation par rapport aux $f_i(\mathbf{x}_k \mathbf{0}, t_{n0})$, puisque l'on a utilisé des éléments discrets dans 2.34 :

$$\frac{\partial R_j(\mathbf{x}_k, t_n)}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = \delta_{ji} \delta_{\mathbf{x}_k + \mathbf{c}_i \Delta t, \mathbf{x}_{k0}} \delta_{n0, n+1} - (1 - \frac{1}{\tau}) \delta_{ji} \delta_{\mathbf{x}_k, \mathbf{x}_{k0}} \delta_{n0, n} - \frac{1}{\tau} \frac{\partial f_j^{eq}}{\partial f_i} \delta_{\mathbf{x}_k, \mathbf{x}_{k0}} \delta_{n0, n+1} \quad (2.35)$$

L'équation 2.34 devient alors :

$$\frac{\partial L}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = \frac{\partial I_0}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} + f_i^*(\mathbf{x}_{k0} - \mathbf{c}_i \Delta t, t_{n0}) - (1 - \frac{1}{\tau}) f_i^*(\mathbf{x}_{k0}, t_{n0+1}) - \frac{1}{\tau} \sum_j f_j^*(\mathbf{x}_{k0}, t_{n0+1}) \frac{\partial f_j^{eq}}{\partial f_i}(\mathbf{x}_{k0}, t_{n0}) = 0 \quad (2.36)$$

On définit alors la fonction d'équilibre adjointe comme :

$$f_i^{*,eq}(\mathbf{x}_{k0}, t_{n0+1}) = \sum_j f_j^*(\mathbf{x}_{k0}, t_{n0+1}) \frac{\partial f_j^{eq}}{\partial f_i}(\mathbf{x}_{k0}, t_{n0}) \quad (2.37)$$

L'équation adjointe s'écrit alors :

$$f_i^*(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t) = f_i^*(\mathbf{x}, t) - \frac{1}{\tau} [f_i^*(\mathbf{x}, t) - f_i^{*,eq}(\mathbf{x}, t)] - \frac{\partial I_0}{\partial f_i(\mathbf{x}, t)} \quad (2.38)$$

On remarque que cette équation est très similaire à l'équation de la LBM classique, à l'exception que les particules se propagent en sens opposé ($-\mathbf{c}_i \Delta t$). Il existe aussi un terme supplémentaire, qui dépend de la fonction coût étudiée. Il est alors possible d'utiliser les mêmes schémas numériques pour résoudre l'équation adjointe que ceux utilisés pour la LBM, ce qui rend cette méthode très attractive.

Il convient maintenant de détailler l'expression de $f_i^{*,eq}(\mathbf{x}, t)$. Pour cela, il est nécessaire d'établir les relations suivantes, issues de la dérivation analytique de 2.22, 2.23 et 2.26 (en posant $\mathbf{j} = \rho \mathbf{u}$) :

$$\frac{\partial \rho}{\partial f_i} = 1 \quad (2.39)$$

$$\frac{\partial \mathbf{j}}{\partial f_i} = \mathbf{c}_i \quad (2.40)$$

$$\frac{\partial \mathbf{u}}{\partial f_i} = \frac{1}{\rho} (\mathbf{c}_i - \mathbf{u}) \quad (2.41)$$

$$\frac{\partial f_j^{eq}}{\partial f_i} = \omega_j (1 + \frac{\mathbf{c}_j \cdot \mathbf{c}_i}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{c}_i \mathbf{u} + \mathbf{u} \mathbf{c}_i - \mathbf{u} \otimes \mathbf{u})) \quad (2.42)$$

Pour rappel, $\mathcal{H}_j^{(2)}$ est le polynôme de Hermite du second ordre qui s'écrit : $\mathcal{H}_{j\alpha\beta}^{(2)} = c_{j\alpha}c_{j\beta} - c_s^2\delta_{\alpha\beta}$. Nous avons alors :

$$f_i^{*,eq} = \sum_{j=1}^{19} \omega_j f_j^* \left[\left(1 + \frac{\mathbf{c}_j \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{u} \otimes \mathbf{u}) \right) + \left(\mathbf{c}_j + \frac{1}{c_s^2} \mathcal{H}_j^{(2)} \right) \cdot \frac{\mathbf{c}_i - \mathbf{u}}{c_s^2} \right] \quad (2.43)$$

La fonction d'équilibre adjointe peut ainsi s'écrire de la façon suivante :

$$f_i^{*,eq}(\mathbf{x}_i, t) = \rho^* + \frac{(\mathbf{c}_i - \mathbf{u}) \cdot \mathbf{j}^*}{c_s^2} \quad (2.44)$$

en définissant les variables adjointes suivantes :

$$\rho^* = \sum_{j=1}^{19} \omega_j f_j^* \left(1 + \frac{\mathbf{c}_j \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{u} \otimes \mathbf{u}) \right) \quad (2.45)$$

$$\mathbf{j}^* = \sum_{j=1}^{19} \omega_j f_j^* \left(\mathbf{c}_j + \frac{1}{c_s^2} \mathcal{H}_j^{(2)} \mathbf{u} \right) \quad (2.46)$$

On remarque ainsi que calculer un adjoint nécessite d'enregistrer le champ de vitesse direct \mathbf{u} à chaque itération du calcul direct. Cela peut s'avérer très coûteux pour des cas industriels mais nous verrons par la suite comment contourner cet obstacle.

2.2.5 Calcul du gradient

Nous pouvons à présent écrire l'expression du gradient obtenu par la méthode adjointe. En partant de l'équation 2.31, on peut écrire :

$$\nabla I_0 = \frac{\partial I_0}{\partial \alpha} + f^* \frac{\partial R}{\partial \alpha} \quad (2.47)$$

En écrivant l'équation précédente sous forme discrète, nous avons :

$$\nabla I_0 = \frac{\partial I_0}{\partial \alpha} + \sum_{i=0}^{q-1} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} f_i^*(\mathbf{x}_k, t_{n+1}) \frac{\partial R_i(\mathbf{x}_k, t_n)}{\partial \alpha} \quad (2.48)$$

L'expression détaillée du gradient dépend du cas étudié (c'est-à-dire du choix de I_0 , de α et de R) comme nous le verrons par la suite.

On remarque ici que le calcul du gradient est indépendant du nombre de paramètre d'optimisation, ce qui constitue le principal intérêt de la méthode adjointe. Comparativement à un calcul par différences finies, le coût de calcul du gradient reste le même quel que soit le nombre de paramètres d'optimisation. Par exemple, si le maillage surfacique d'un véhicule possède 1 million de mailles et que les paramètres d'optimisation α sont les coordonnées x, y, z de chaque maille, cela fait 3 millions de calculs de gradient par différences finies. L'optimisation par adjoint est très intéressante dans ce cas car le nombre de paramètres d'optimisation est alors très élevé.

Chapitre 3

Application à un problème industriel

Pour conclure ce chapitre de bibliographie, les problématiques liées à un problème industriel sont décrites dans cette partie. En effet, la thèse ayant pour objectif final l'application du solveur adjoint sur un véhicule complet, il faut soulever les complications qui y sont liées.

3.1 Modèles de collision

3.1.1 Opérateur de collision BGK

L'étape de collision est calculée en utilisant l'opérateur de collision Bhatnagar-Gross-Krook [9], présenté dans la sous-section 2.2.3. Celui-ci est bien adapté aux écoulements laminaires, et est simple car seul un temps de relaxation est utilisé pour les fonctions de distribution.

3.1.2 Opérateur de collision DRT

Une idée dans ProLB a été de mélanger le modèle BGK dans lequel tous les modes sont relaxés avec un même temps de relaxation et la collision régularisée qui permet de relaxer de manière sélective les modes physiques. Cet opérateur a deux temps de relaxation pour différencier les moments hydrodynamiques des moments fantômes. En effet, disposer de deux temps de relaxation distincts permet d'atténuer les effets des modes fantômes qui donnent naissance à des instabilités pour des écoulements à nombre de Reynolds élevé. L'idée sous-jacente au modèle DRT est ainsi de relaxer les fonctions de distribution vers l'équilibre en conservant la masse et la quantité de mouvement, en reconstruisant la partie hors-équilibre des fonctions de distributions à travers une régularisation du second ordre inspirée de l'article de Latt et Chopard [46].

Dans leur article, ils rappellent que le développement de Chapman-Enskog au premier ordre en ϵ donne les équations de conservation de la masse et de la quantité de mouvement dans un régime hydrodynamique. Ces équations dépendent seulement des variables hydrodynamiques suivantes : la masse, la vitesse et le tenseur des contraintes. Les lois de l'hydrodynamique dépendent donc de 6 variables en 2D, et de 10 en 3D. Cela signifie que le nombre d'inconnues f_i est surdéterminé (généralement 9 en 2D et 19 en 3D), et Latt et Chopard proposent de les remplacer par un ensemble équivalent d'inconnues différentes menant aux mêmes grandeurs hydrodynamiques. Cela permet de forcer les fonctions de distribution à mieux s'accorder à leurs valeurs hydrodynamiques en $o(\epsilon)$.

On peut aussi noter l'existence d'autres modèles à temps de relaxation multiples, tels que les modèles "de type MRT" (Multiple Relaxation Time), basés sur les travaux de D'Humières [19], ou encore les modèles Cascaded Lattice Boltzmann [25] et Cumulant Lattice Boltzmann [26]. Pour une lecture approfondie des modèles à temps de relaxation multiples, le lecteur est dirigé vers la thèse de Félix Gendre [27] qui décrit notamment le modèle DRT utilisé dans LaBS dans le contexte général des modèles MRT.

L'équation du modèle DRT s'écrit sur les fonctions de distributions :

$$f_i(\mathbf{x}_{k+1}, t_{n+1}) = \left(1 - \frac{1}{\tau^N}\right) f_i(\mathbf{x}_k, t_n) + \frac{1}{\tau^N} f_i^{eq}(\mathbf{x}_k, t_n) + \frac{\tau^S - \tau^N}{\tau^S \tau^N} \frac{\omega_i}{2c_s^4} \sum_i \mathcal{H}_{i,\alpha\beta}^{(2)} P_{\alpha\beta}^{(1)} \quad (3.1)$$

avec :

$$\mathcal{H}_{i,\alpha\beta}^{(2)} = \mathbf{c}_{i,\alpha} \mathbf{c}_{i,\beta} - c_s^2 \delta_{\alpha\beta} \quad (3.2)$$

et

$$P_{\alpha\beta}^{(1)} = \sum_i \mathcal{H}_{i,\alpha\beta}^{(2)} (f_i - f_i^{eq}) \quad (3.3)$$

avec $\mathcal{H}_{i,\alpha\beta}^{(2)}$ le polynôme de Hermite du second ordre, $P_{\alpha\beta}^{(1)}$ la partie hors équilibre du moment d'ordre 2, τ^S le temps de relaxation physique et τ^N un temps de relaxation numérique utilisé pour atténuer les modes fantômes.

3.2 Conditions limites de paroi

L'algorithme de paroi choisi pour modéliser l'écoulement proche de la paroi est très important, notamment dans les écoulements turbulents, car il doit refléter précisément les phénomènes physiques qui s'y produisent (formation d'une couche limite, décollement, etc.). En effet, la localisation précise de la zone de décollement par exemple sur un véhicule permet de mieux contrôler les pertes énergétiques qui en découlent, il est donc nécessaire de développer des modèles précis permettant de reproduire ces phénomènes. La problématique des algorithmes de paroi en LBM peut se résumer de la façon suivante

grâce à la figure 3.1 : comment calculer f_7 par exemple, sachant que les nœuds et fonctions de distributions se trouvant dans le domaine solide sont inconnues .

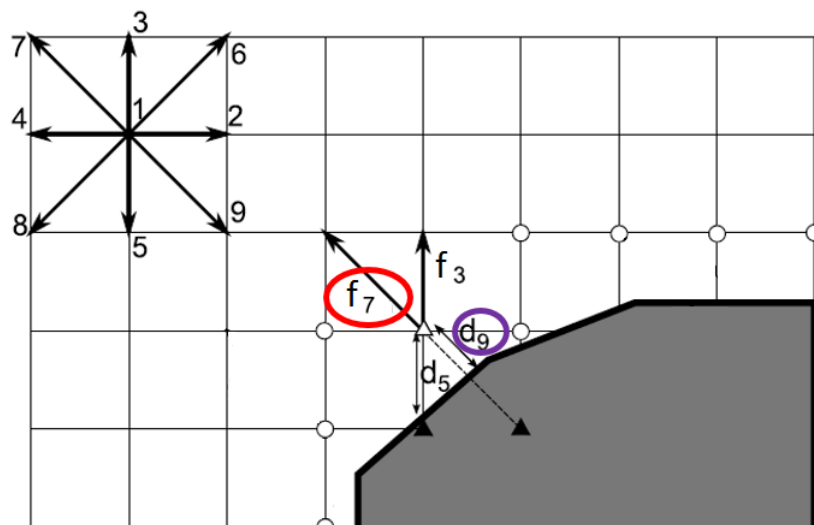


FIGURE 3.1 – Problématique à la paroi

Dans notre étude, nous avons adopté plusieurs approches, de la plus simple à la plus complexe. Nous détaillerons trois approches de modélisation proche paroi, par ordre de complexité.

3.2.1 Condition de rebond : Bounce-Back

La condition de link bounce-back est présentée dans un premier temps, car elle est relativement simple à implémenter. Sa formulation ne nécessite pas de connaître la distance à la paroi. L'algorithme Link Bounce-Back est basée sur une opération de collision standard, éventuellement modifiée par un terme d'effort externe permettant de modéliser une vitesse de paroi non nulle et sur le retournement des fonctions post-collision "pendant la propagation". Cet algorithme est d'ordre 1 pour l'estimation du champ de vitesse [34] et il est d'ordre 0 pour le champ de pression (l'erreur sur le champ de pression ne dépend pas de la taille de la maille). Pour le cas très particulier où le nœud de frontière se trouve exactement à mi-chemin entre la frontière solide et le premier nœuds fluide, le modèle Link Bounce-Back a une convergence à l'ordre 2 en vitesse et 1 en pression.

$$f_j(\mathbf{x}_k, t_{n+1}) = f_{opp(j)}^{col}(\mathbf{x}_k, t_n) \quad (3.4)$$

avec la fonction de collision définie de la façon suivante :

$$f_{opp(j)}^{col}(\mathbf{x}_k, t) = f_{opp(j)}(\mathbf{x}_k, t_n) - \frac{1}{\tau} [f_{opp(j)}(\mathbf{x}_k, t_n) - f_{opp(j)}^{eq}(\mathbf{x}_k, t_n)] \quad (3.5)$$

3.2.2 Interpolation de Ginzburg

L'interpolation de Ginzburg [30] repose sur une interpolation du modèle Link Bounce Back, et est basée sur le calcul de la distance d à la paroi. Ce modèle est donc plus précis que le précédent, puisqu'il est d'ordre 2 en vitesse et 1 en pression :

$$f_j(\mathbf{x}_k, t_{n+1}) = f_{opp(j)}^{col}(\mathbf{x}_k, t_n) + \frac{1 - 2d_{opp(j)}}{1 + 2d_{opp(j)}} [f_{opp(j)}^{col}(\mathbf{x}_k + \mathbf{c}_j \Delta t, t_n) - f_j^{col}(\mathbf{x}_k, t_n)] \quad (3.6)$$

avec $d_{opp(j)} = d_g$ par exemple dans la figure 3.1. La distance utilisée $d \in [0, 1]$ est un paramètre sans dimension permettant de connaître la position relative du nœud fluide de frontière x_f par rapport au nœud solide sur la paroi x_w :

$$d = \frac{|x_f - x_w|}{|x_f - x_b|} \quad (3.7)$$

Le schéma 3.2 permet de mieux comprendre les notations utilisées :

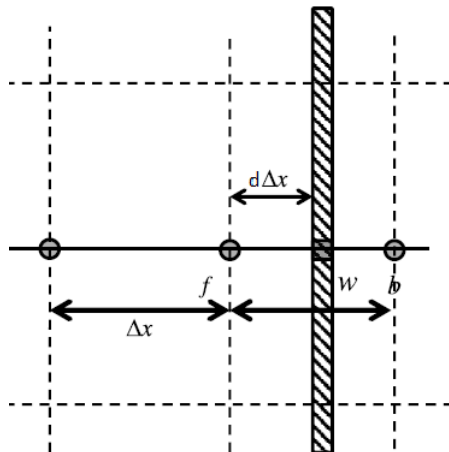


FIGURE 3.2 – distance à la paroi

3.2.3 Méthode de reconstruction utilisée dans ProLB

Le modèle de frontière solide immergée utilisant la méthode de reconstruction des fonctions de distribution est basée sur une décomposition formelle de la fonction de distribution en une partie à l'équilibre f^{eq} et une partie hors équilibre f^{neq} :

$$f_j(\mathbf{x}_k, t_n) = f_j^{eq}(\mathbf{x}_k, t_n) + (f_j(\mathbf{x}_k, t_n) - f_j^{eq}(\mathbf{x}_k, t_n)) = f_j^{eq}(\mathbf{x}_k, t_n) + f_j^{neq}(\mathbf{x}_k, t_n) \quad (3.8)$$

On peut montrer par le développement de Chapman-Enskog[[93], [83]] qu'on a la relation :

$$f_j^{neq}(\mathbf{x}_k, t_n) = -\tau_g \rho \frac{\omega_j}{c_s^2} \sum_{\alpha, \beta} (c_{j,\alpha} c_{j,\beta} - c_s^2 \delta_{\alpha\beta}) S_{\alpha\beta} \quad (3.9)$$

avec

$$S_{\alpha\beta} = \frac{1}{2} \left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \quad (3.10)$$

Etant donné que la fonction d'équilibre f^{eq} s'exprime en fonction des variables macroscopiques, on constate alors que les expressions précédentes permettent d'exprimer de manière explicite les fonctions de distribution à partir des variables macroscopiques et de leurs dérivées spatiales. Ce modèle de définition des conditions aux limites a déjà été appliqué au cas de parois planes coincidentes avec les nœuds de frontière[[29], [32], [79], [80]].

Dans ProLB, le choix a été fait de passer par la formulation utilisant le tenseur des déformations S_{ij} . Les approches basées sur la méthode de reconstruction proposée depuis le début des années 90 sont cependant restreintes au cas d'une paroi plane, avec des nœuds de calcul qui coïncident avec la paroi. A la connaissance de l'auteur, le premier article permettant de généraliser cette approche au cas des parois quelconques a été publié en 2010 par Verschaeve et Muller [91]. Dans le cadre de ProLB une approche assez similaire pour étendre la méthode de reconstruction au cas plus général des frontières solides immergées a été développée. L'idée du modèle de reconstruction de ProLB est de calculer aux nœuds de frontière toutes les variables macroscopiques par interpolation/extrapolation, ce qui permet de redéfinir la fonction d'équilibre, puis les S_{ij} sont évaluées par différences finies afin d'évaluer la partie hors équilibre de la fonction de distribution.

3.3 Raffinement de maillage

Pour un cas industriel, l'utilisation de raffinements de maillage est indispensable pour trouver le bon compromis entre précision à la paroi, et donc taille de maille petite, et coût de calcul, et donc taille de maille grossière loin de la paroi. La modélisation des écoulements turbulents nécessite notamment une taille de maille suffisamment petite pour pouvoir simuler les phénomènes qui se produisent proche de la paroi. Le maillage ainsi obtenu est homogène par bloc : dans chaque bloc, l'algorithme Boltzmann sur Réseau standard peut être appliqué, et l'on divise la taille de maille par 2 d'un bloc à l'autre. Le choix a été fait dans le solveur de stocker les données aux coins des mailles cubiques comme le montre la figure 3.3. Ce choix permet de minimiser le nombre de nœuds problématiques : au niveau de la transition, un nœud sur deux seulement n'est pas coïncident spatialement. Le maillage est donc non conforme au niveau de la transition : certains nœuds ne possèdent pas tous leurs voisins naturels.

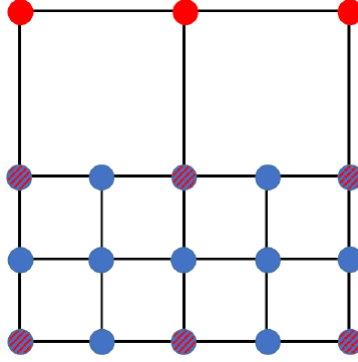


FIGURE 3.3 – Schéma d'un raffinement de maillage en 2D

Un problème qui se pose concerne la coïncidence temporelle. En LBM, le rapport dx/dt (pas d'espace / pas de temps) doit être maintenu constant. Ainsi, si on suppose que dans une zone dite fine le calcul est fait aux itérations $t, t + 1, t + 2, t + 3, t + 4, \dots$ alors, pour la région de maillage dite grossière adjacente le calcul se fera deux fois moins souvent soit aux itérations $t, t + 2, t + 4, \dots$. Ainsi, au niveau des nœuds faisant partie de l'interface de raccordement des maillages, la coïncidence temporelle des données n'aura lieu qu'un pas de temps sur deux. Pour résoudre ce problème, une interpolation temporelle est effectuée. On utilise une interpolation polynomiale d'ordre 3 :

$$f_i^f(\mathbf{x}, t) = \alpha f_i^f(\mathbf{x}, t - 2) + \beta f_i^f(\mathbf{x}, t - 1) + \gamma f_i^f(\mathbf{x}, t + 1) \quad (3.11)$$

On note f (fine) la zone de maillage fin et c (coarse) la zone de maillage grossier. La maille grossière de l'interface (maille commune aux deux maillages au niveau de l'interface) est appelée Transition Coarse. Dans l'autre sens, le passage du maillage grossier vers le maillage fin se fait au niveau de la maille commune au deux maillages, appelée Transition Fine. Ce nœud reçoit donc l'information du nœud Transition Coarse qui est co-localisé. Les variables macroscopiques ρ, u et donc les fonctions d'équilibre f^{eq} sont continues au travers d'une interface entre deux maillages de taille différente. Par contre, les fonctions de distribution ne sont pas continues au travers d'une interface entre deux maillages de taille différente. Ces fonctions sont reliées entre elles par un recalage de variables ([23], [22] et [50]) :

$$f_i^f(\mathbf{x}, t) = \frac{1}{2\tau_f + 1} ((2\tau_f) f_i^c(\mathbf{x}, t) + f_i^{eq}(\mathbf{x}, t)) \quad (3.12)$$

$$f_i^c(\mathbf{x}, t) = \frac{1}{2\tau_f} ((2\tau_f + 1) f_i^f(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \quad (3.13)$$

Comme expliqué précédemment, les valeurs des variables macroscopiques ne dépendent pas de la taille de la maille. Par conséquent, la fonction d'équilibre non plus : $f^{eq,f} = f^{eq,c} = f^{eq}$

Enfin, les nœuds qui se trouvent au milieu d'une arête ou d'une face d'une interface entre deux maillages de taille différentes sont appelés Interspace Edge et Interspace Face. Le calcul se fait d'abord sur les nœuds Interspace Edge puis sur les nœuds Interspace Face avec les schémas suivants :

- Les nœuds Interspace Edge sont interpolés avec un schéma 1D, en prenant les points Transition Fine se situant sur la même arête qu'eux, à une distance dx et $3dx$ de chaque côté.
- Puis les nœuds Interspace Face sont interpolés avec un schéma 2D, en prenant les nœuds Interspace Edge précédemment interpolés et se situant sur la même face qu'eux, à une distance dx et $3dx$ dans les quatre directions du plan de l'interface.

Dans les situations d'écoulement turbulent, il est apparu nécessaire d'améliorer l'algorithme de gestion des transitions entre grilles afin d'assurer une meilleure continuité des tailles des structures turbulentes qui sont simulées de part et d'autre de l'interface. Pour cela un algorithme de restriction de l'opérateur de collision, côté maillage fin avant le transfert de l'information vers le maillage grossier, mais qui ne sera pas détaillé ici, a été mis en place dans [87].

3.4 Modèle de turbulence

Trois grandes familles de modèles de turbulence existent à ce jour : les modèles RANS, les modèles LES et l'approche de modélisation explicite DNS. Leurs degrés de modélisation et de résolution sont différents, selon la précision souhaitée. Alors que les modèles DNS résolvent toutes les échelles de la turbulence, les modèles RANS les modélisent toutes, et les modèles LES constituent un intermédiaire entre les deux en résolvant les grandes structures turbulentes et en modélisant les plus petites comme le montre la figure 3.4 :

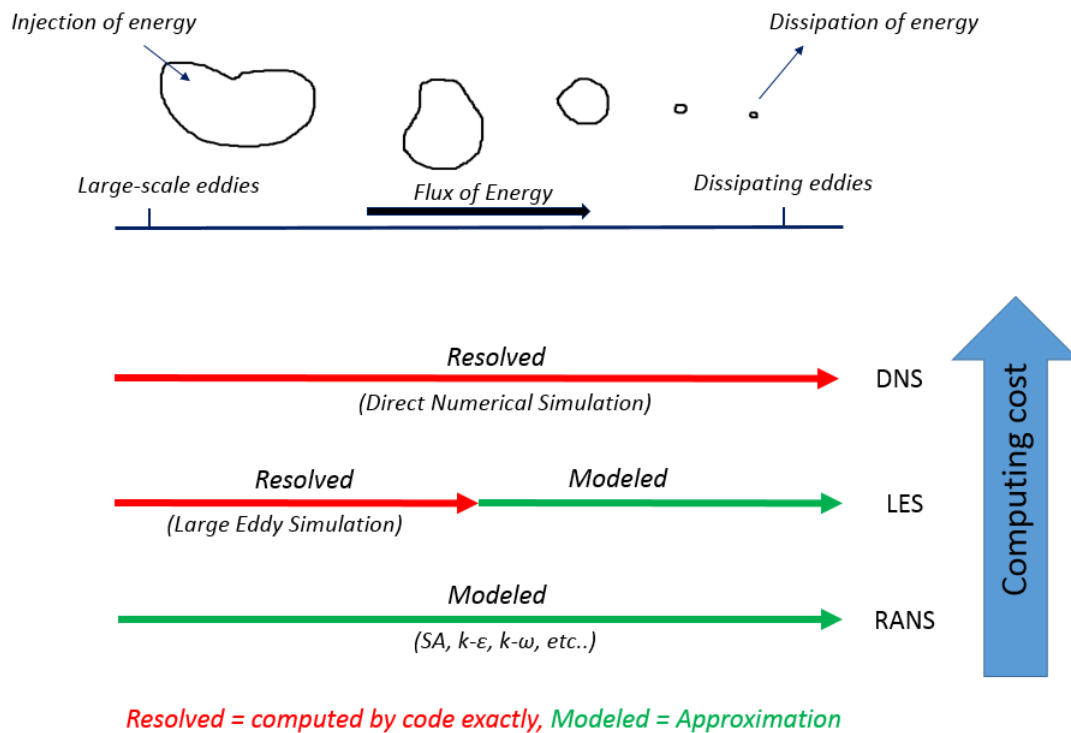


FIGURE 3.4 – Modèles de turbulence

Ici, on choisit d'utiliser un modèle LES (ou modèle de sous-maille) afin de prendre en compte les échelles non résolues dans la couche limite, et plus précisément une variante du modèle de Smagorinsky nommée le "shear-improved Smagorinsky" proposé par [48]. Dans ce modèle, on calcule la viscosité turbulente de la façon suivante :

$$\nu_t = (C_S^2 \Delta x)^2 (|S| - \mathcal{S}) \quad (3.14)$$

avec C_S la constante de Smagorinsky, Δx le pas de maillage local, S le tenseur des déformations et $|S|$ la norme de ce dernier. Le terme correctif \mathcal{S} est la norme du filtre basse fréquence de S qui permet de réduire la dissipation excessive introduite par le modèle de Smagorinsky dans les écoulements fortement cisailés, en retranchant l'effet du cisaillement moyen.

Dans la LBM, la viscosité turbulente est prise en compte dans le calcul du temps de relaxation physique de la façon suivante :

$$\tau^S = \frac{\nu + \nu_t}{c_s^2} + \frac{1}{2} \quad (3.15)$$

Pour la thèse, l'hypothèse de turbulence gelée sera effectuée pour le calcul adjoint. Cela signifie que la viscosité turbulente sera la même pour le calcul direct et le calcul adjoint. Il n'y aura donc pas de modèle de turbulence implémenté dans le code adjoint

puisqu'on récupère directement le terme de viscosité turbulente du résultat du calcul direct, d'où le terme de "turbulence gelée". Cela simplifie grandement l'implémentation de l'adjoint, mais dégrade la précision du gradient. Différents auteurs ont aussi effectué cette approximation, et ont obtenu des résultats concluants [68, 71].

3.5 Approximation temporelle du problème direct

Sachant que tous les pas de temps du calcul direct doivent être enregistrés pour pouvoir faire tourner le calcul adjoint, la nature stationnaire ou instationnaire du calcul direct est primordiale. En effet, si le calcul est stationnaire, on peut n'enregistrer que le champ convergé correspondant à la dernière itération, et le solveur adjoint utilisera ce champ à chaque itération jusque attendre lui-même un état convergé. Au contraire, si le calcul direct est instationnaire, théoriquement il est nécessaire d'enregistrer tous les pas du temps du calcul pour avoir accès aux variables macroscopiques (p et \mathbf{u} dans les équations 2.38 et 2.44), ce qui peut s'avérer très lourd pour des cas industriels turbulents et instationnaires à nombre de Reynolds élevé. Pour remédier à ce problème, il existe plusieurs approches qui seront abordées dans cette section.

3.5.1 Direct stationnaire-Adjoint stationnaire

Pour juger de l'état convergé d'un calcul stationnaire, Yaji et al. ([94]) imposent un critère sur la continuité de l'écoulement d'un pas de temps au suivant. Si l'écoulement ne varie plus ou très peu, ils considèrent que le calcul est convergé. Pour cela, ils utilisent le critère suivant :

$$\left| \frac{u^N - u^{N-1}}{u_N} \right| < \epsilon_u \quad (3.16)$$

avec ϵ_u le critère de condition stationnaire pour la vitesse du fluide, imposée à : $\epsilon_u = 1e^{-4}$. Ils utilisent alors les états convergés du calcul direct et du calcul adjoint pour calculer les sensibilités.

3.5.2 Direct quasi instationnaire - Adjoint stationnaire

Norgaard et al. [64] considèrent quant à eux un problème instationnaire, qui nécessite d'enregistrer le champ direct en entier (c'est-à-dire à chaque itération du calcul). Leur méthode s'applique à un problème 2D relativement peu coûteux, ils admettent cependant que pour un problème industriel en 3D cela s'avère trop coûteux. La solution qu'ils proposent alors est d'implémenter un algorithme de "checkpointing" pour enregistrer le calcul direct non pas à toutes les itérations mais seulement à certains intervalles "d'échantillonnage" (ou checkpoints). Cela permet de faire tourner un calcul adjoint en relisant les données des "checkpoints" au cours du calcul, afin de calculer le gradient grâce à la connaissance des champs direct et adjoint sur ces différents intervalles et non sur toutes les itérations. La taille mémoire nécessaire à l'enregistrement du champ direct

est alors réduite. Pour une bibliographie détaillée de ce genre de méthodes, le lecteur est dirigé vers [15].

3.5.3 Direct instationnaire - Adjoint instationnaire

L'approche "direct instationnaire - adjoint instationnaire" consiste à enregistrer tous les pas de temps du champ direct, puis à les utiliser en temps inverse pour faire tourner le calcul adjoint. Le gradient nécessite d'avoir accès aux données du champ direct et adjoint sur toutes ces itérations, comme nous le verrons dans la section 4.1. Cette approche a été utilisée par différents auteurs : [55], [85], [90], [64], mais elle est limitée pour l'instant à des cas 2D car elle est trop lourde en stockage mémoire et temps de relecture du champ direct à chaque itération par l'adjoint pour des cas 3D. Le problème posé par cette approche n'est pas un temps de calcul long dû à de nombreuses opérations arithmétiques mais est uniquement dû à un problème de stockage de données, de taille de fichier et de temps de lecture par le solveur adjoint.

3.5.4 Comparaison des méthodes

Nadarajah et Jameson [60] ont étudié la question de la stationnarité du problème direct en détail et l'ont appliqué à un écoulement autour d'un profil d'aile simulé avec les équations d'Euler. Ils ont comparé plusieurs méthodes :

- la méthode complètement instationnaire (qu'ils appellent "full unsteady design" soit "unsteady-flow unsteady-adjoint"),
- la méthode partiellement instationnaire ("unsteady-flow steady-adjoint"),
- et la méthode stationnaire dont le problème direct est moyenné en temps ("time-averaged-flow steady-adjoint design").

La première méthode ("full unsteady design method") correspond à ce qui a été présenté dans la sous-section 3.5.3 et consiste à enregistrer tous les pas de temps du problème direct pour calculer un adjoint instationnaire.

La deuxième méthode ("partial unsteady design") qui correspond à la sous-section 3.5.2 est basée sur le calcul instationnaire complet du problème direct, dont on enregistre des étapes intermédiaires. Cela permet de calculer un adjoint stationnaire à partir des étapes enregistrées.

Enfin la dernière méthode ("time-averaged-flow steady-adjoint design") implique de calculer une solution moyennée en temps du champ direct pour calculer un seul adjoint stationnaire. Cette dernière méthode mène à l'hypothèse suivante : le calcul d'un adjoint stationnaire basé sur la moyenne temporelle de la solution directe donne des sensibilités correctes.

Nadarajah et Jameson ont trouvé que pour le problème qu'ils étudient, le calcul des gradients est très similaire quel que soit la méthode choisie. Pour des écoulements

très turbulents autour de véhicules, plusieurs auteurs [41], [68],[71] ont aussi obtenu des résultats concluants en utilisant cette hypothèse (utiliser une solution directe moyennée pour calculer un adjoint stationnaire). Nous utiliserons donc cette même hypothèse ici car nos cas d'applications industriels sont similaires à ceux mentionnés précédemment, et cette méthode est la moins coûteuse des trois en termes d'espace mémoire et temps de calcul. Nous comparerons aussi cette hypothèse à la méthode "full unsteady design" pour un cas d'étude 2D dans le chapitre 6.

Conclusion de la partie I

Cette première partie bibliographique a permis de passer en revue les différentes méthodes d'optimisation existantes, ainsi que leurs applications à des problèmes industriels complexes en mécanique des fluides.

Le chapitre 1 a montré trois types d'optimisation : paramétrique, surfacique et topologique. Une explication de chaque type d'optimisation a été donnée, afin de cibler la plus adaptée à notre étude. Sachant que l'aérodynamique externe est très contrainte par le design du véhicule, l'optimisation surfacique a été choisie car elle permet de conserver la forme globale du véhicule sans créer de trous dans la forme, contrairement à l'optimisation topologique. Ensuite, les étapes nécessaires à la réalisation d'une boucle d'optimisation complète ont été détaillées. Une liste non exhaustive de fonctions coûts caractérisant un écoulement en aérodynamique interne puis externe a été donnée, puis différentes façons de calculer les gradients ont été détaillées. Ici, la méthode adjointe a été choisie pour le calcul des gradients surfaciques car elle est très intéressante en termes de coût de calcul. Enfin, plusieurs méthodes de descente du gradient ont été présentées. La méthode de descente du gradient à pas constant est choisie pour modifier la géométrie, car elle est la plus simple à implémenter.

Le chapitre 2 a ensuite montré l'étude bibliographique réalisée sur la méthode adjointe appliquée aux équations de la mécanique des fluides. Une étude reposant sur la méthode adjointe avec les équations de Navier-Stokes a été réalisée (qui n'est pas montrée ici pour des raisons de confidentialité) et a permis de montrer que des solveurs existent déjà et sont fonctionnels pour l'aérodynamique interne. Il a ainsi été choisi de se focaliser sur l'optimisation de l'aérodynamique externe, qui est un sujet ayant plus de potentiel, tant sur le plan académique qu'industriel. Une étude bibliographique a aussi été réalisée sur la méthode adjointe appliquée aux équations de Lattice-Boltzmann. Celle-ci a montré qu'il n'existe pas à ce jour de solveur adjoint industriel pour les équations de Lattice-Boltzmann (qui permettent de simuler l'aérodynamique externe chez Renault), contrairement aux équations de Navier-Stokes.

Enfin, le chapitre 3 a montré certaines fonctionnalités de ProLB, le solveur basé sur les équations de Lattice-Boltzmann et utilisé chez Renault pour simuler l'aérodynamique externe. Le développement du solveur adjoint devra ainsi prendre en compte toutes les

fonctionnalités de ProLB nécessaires à notre étude : un modèle de collision à deux temps de relaxation pour les écoulements turbulents, des algorithmes pour les raffinements de maillage, un algorithme de paroi, une modélisation de la turbulence, et enfin la dépendance temporelle de l'écoulement.

En résumé, la thèse se concentrera sur le développement d'un solveur adjoint, appliqué à l'aérodynamique externe, simulée avec la méthode de Lattice-Boltzmann. Ce solveur adjoint permettra de calculer les sensibilités sur chaque nœud surfacique du maillage du véhicule. Une étape de lissage sera ensuite nécessaire pour lisser ces sensibilités, puis un morphing direct sera appliquée afin de pouvoir déformer le véhicule. La démonstration d'une boucle complète d'optimisation sera réalisée, avec un calcul de vérification sur la géométrie optimisée qui montrera que les forces aérodynamiques qui nous intéressent sont bien réduites.

Le développement du solveur adjoint étant un processus complexe, chaque étape de son développement sera explicité dans le manuscrit, en suivant la chronologie de la thèse. La partie II se concentrera sur le développement du solveur adjoint sur des cas simples 2D, et la partie III montrera son application et les développements associés pour appliquer le solveur adjoint sur un cas industriel complexe, objectif final de la thèse. Ce cas complexe sera un écoulement d'air à vitesse élevée, en régime turbulent et instationnaire, sur un véhicule réaliste, avec un maillage raffiné 3D.

Deuxième partie

**Développement des algorithmes
LBM adjoints et validation sur
des cas 2D académiques**

Cette partie présente le développement du solveur adjoint sur des cas 2D académiques, relativement simples. Il est important de noter que le modèle D3Q19 est utilisé même pour les cas 2D présentés dans cette partie, qui sont en réalité des cas 2D extrudés, avec 1 couche de maille volumique. Ceci permet de simplifier la démarche et de garder la même quadrature lors du passage à des cas 3D.

Pour cela, le schéma 3.5 montre la démarche suivie : la partie de gauche du schéma montre que le solveur ProLB contient un certain nombre de fonctionnalités. Seules celles nécessaires à nos cas d'étude seront étudiées ici (celles entourées en trait plein en bleu), à savoir les équations régissant le comportement du fluide en cœur de fluide (choix du modèle de collision), en proche paroi (choix de l'algorithme de paroi) et au niveau d'une transition de maillage (choix des algorithmes des raffinements de maillage). Les fonctionnalités entourées en trait pointillé en bleu sont celles sur lesquelles nous reviendrons plus tard au cours de la thèse, pour les cas industriels et turbulents.

La suite de la démarche consiste à utiliser ces algorithmes pour étudier un cas test aérodynamique, pour lequel d'autres questions apparaissent. La partie à droite du schéma montre donc les choix à effectuer, c'est-à-dire la façon de calculer la force de traînée F_x , et comment traiter le problème selon qu'il est stationnaire ou instationnaire.

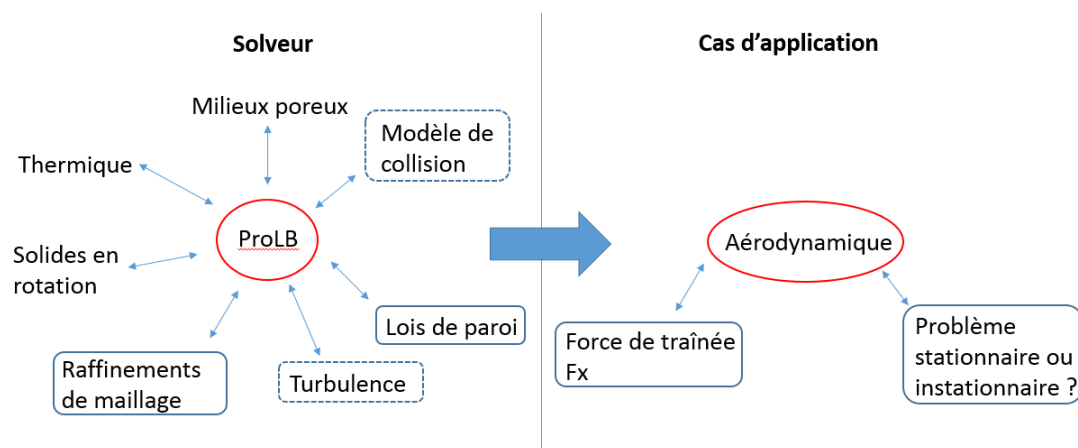


FIGURE 3.5 – Elements nécessaires à l'étude d'un cas aérodynamique en LBM avec ProLB

Nous verrons comment ces choix influencent le développement de l'adjoint, en termes de complexité de la méthode mais aussi de précision des résultats. Le chapitre 4 montrera ainsi les cas d'étude ayant permis de comparer différents algorithmes de paroi ainsi que les algorithmes de raffinement de maillage. Le chapitre 5 nous détaillera ensuite les résultats obtenus en utilisant deux formulations différentes pour la force de traînée. Enfin, le chapitre 6 montrera les résultats obtenus en comparant deux approximations temporelles possibles d'un écoulement instationnaire. Le choix du modèle de collision

n'est pas étudié ici car le modèle BGK est adapté aux écoulements laminaires, et nous nous intéressons seulement à ce type d'écoulement dans cette partie. Nous n'utilisons pas non plus de modèle de turbulence dans cette partie car elle concerne seulement des écoulements laminaires.

Concernant les conditions limites utilisés pour les cas étudiés dans cette partie, nous utilisons les suivantes :

En entrée d'un domaine fluide, il est usuel de définir une vitesse d'écoulement fixe et une pression libre. En LBM, cela se traduit par une mise à l'équilibre des fonctions de distributions, en fixant la vitesse et en interpolant la densité par les points environnants. En sortie, on fixe généralement la pression et la vitesse est libre. Cela se traduit en LBM de la même façon par une mise à l'équilibre des fonctions de distribution, mais cette fois on fixe la densité et on interpole la vitesse.

Chapitre 4

Choix de l’algorithme de paroi et des raffinements de maillage

Le choix de l’algorithme de paroi est une étape importante dans le développement d’un solveur adjoint, puisqu’elle intervient dans le calcul du gradient. Il faut donc la choisir en prenant en compte plusieurs paramètres : sa simplicité d’implémentation, le niveau de difficulté pour trouver sa formulation adjointe, ainsi que la précision attendue dans les résultats obtenus. Ce chapitre permet donc de comparer plusieurs algorithmes de paroi qui ont été étudiés, afin de mettre en évidence le raisonnement suivi au cours du doctorat. Les avantages et inconvénients de chaque algorithme de paroi seront montrés pour finalement choisir l’algorithme de paroi le plus adapté à nos problèmes.

4.1 Etude préliminaire

Cette première étude nous permet de développer et de coder dans le logiciel les premiers éléments du solveur adjoint, à savoir l’implémentation du modèle de collision BGK adjoint, et du calcul du gradient par l’adjoint. Nous allons étudier un problème d’optimisation qui a déjà été documenté dans la littérature afin de vérifier la validité de notre code, pour partir sur des bases solides.

4.1.1 Contrôle actif en acoustique : diminution du bruit

Problème direct

Le cas test étudié ici reprend celui de [90], afin de vérifier dans un premier temps que l’implémentation de l’algorithme LBM adjoint et du gradient dans ProLB est correcte. L’objectif est de réduire le bruit sur une ligne Γ dû à une source acoustique proche. La géométrie du problème est une grille de 200 x 100 nœuds espacés de 2 cm, soit une grille de 4 x 2 m^2 . En prenant l’origine (0,0) de la géométrie au coin inférieur gauche, les coordonnées de la source acoustique x_{ks} sont (3;1.4). La ligne Γ s’étend de (2;0) à (2;2), et on positionne une source de contrôle x_{kc} à (1;1.4), permettant de contrôler le

bruit sur Γ et symétrique à la source x_{ks} par rapport à Γ . Le problème est représenté sur le schéma suivant :

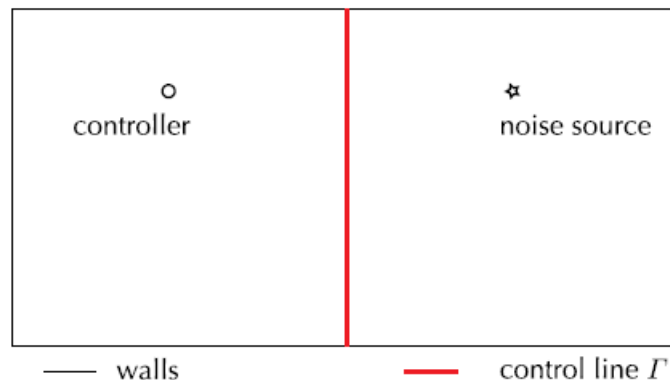


FIGURE 4.1 – Schéma du cas test acoustique

Le paramètre d'optimisation est l'amplitude du signal émis par la source de contrôle x_{kc} . Ce cas test représente un cas simplifié de contrôle acoustique ayant pour objectif la réduction du bruit acoustique. La suite de cette section présente la mise en données du problème, ainsi que les équations détaillées pour ce cas spécifique. Enfin les résultats obtenus seront présentés, ainsi qu'une conclusion sur la validité du modèle adjoint pour ce cas test.

Les paramètres suivants sont choisis pour la simulation : la densité ρ_0 est de 1 kg/m^3 , la viscosité de $0.05 \text{ m}^2/\text{s}$, la vitesse du son de 57.74 m/s , et l'amplitude de la source x_{ks} est fixée à ρ_0 . La pulsation Ω de la source x_{ks} est de 3000 rad/s . Les sources acoustiques x_{ks} et x_{kc} sont introduites comme des sources ponctuelles de masse dans les équations directes de la LBM :

$$R_i = f_i(\mathbf{x}_k + \mathbf{c}_i \Delta t, t_{n+1}) - f_i(\mathbf{x}_k, t_n) + \frac{1}{\tau} [f_i(\mathbf{x}_k, t_n) - f_i^{eq}(\mathbf{x}_k, t_n)] + \omega_i \rho_0 \delta_{\mathbf{x}_k, \mathbf{x}_{ks}} \sin(\Omega t + \pi) + \omega_i \alpha \delta_{\mathbf{x}_k, \mathbf{x}_{kc}} \sin(\Omega t) = 0 \quad (4.1)$$

avec α le paramètre d'optimisation et $\rho_0 = 1$.

Le pas de temps de 0.0002 s pour cette simulation, et l'on effectue 200 pas de temps ce qui donne un temps total noté t_f de 0.04 s . Concernant les conditions limites, des conditions périodiques sont utilisées pour tous les bords du domaine. Pour le calcul adjoint, les conditions limites sont aussi périodiques.

Problème adjoint

La fonction coût utilisée ici est le niveau de pression moyenne quadratique sur la durée du calcul total et sur la ligne Γ :

$$I_0 = \sum_{k \in \Gamma} \sum_{n \in T} \frac{1}{2} p^2(\mathbf{x}_k, t_n) d\Gamma dt \quad (4.2)$$

Sachant que $p = \rho c_s^2$ en LBM et que $\rho = \sum_i f_i$ on peut écrire :

$$I_0 = \sum_{k \in \Gamma} \sum_{n \in T} \frac{1}{2} \rho^2(\mathbf{x}_k, t_n) c_s^4 d\Gamma dt = \sum_{k \in \Gamma} \sum_{n \in T} \frac{1}{2} \left(\sum_i f_i \right)^2(\mathbf{x}_k, t_n) c_s^4 d\Gamma dt \quad (4.3)$$

On rappelle que (sous-section 2.2.4) :

$$\frac{\partial \rho}{\partial f_i} = 1 \quad (4.4)$$

On peut calculer le terme source qui apparait dans l'adjoint :

- si $k \in \Gamma$:

$$\frac{\partial I_0}{\partial f_i} = c_s^4 \rho \quad (4.5)$$

- sinon :

$$\frac{\partial I_0}{\partial f_i} = 0 \quad (4.6)$$

Pour résoudre le champ adjoint (équation 2.38), on utilise un algorithme très similaire à celui de la LBM classique avec une étape de collision et une étape de propagation :

- si $k \in \Gamma$:

$$f_i^{*,col}(\mathbf{x}_k, t_n) = f_i^*(\mathbf{x}_k, t_n) - \frac{1}{\tau} [f_i^*(\mathbf{x}_k, t_n) - f_i^{*,eq}(\mathbf{x}_k, t_n)] - (c_s^4 \rho)_{\mathbf{x}_k \in \Gamma} \quad (4.7)$$

$$f_i^*(\mathbf{x}_k - \mathbf{c}_i \Delta t, t_{n-1}) = f_i^{*,col}(\mathbf{x}_k, t_n) \quad (4.8)$$

- sinon :

$$f_i^{*,col}(\mathbf{x}_k, t_n) = f_i^*(\mathbf{x}_k, t_n) - \frac{1}{\tau} [f_i^*(\mathbf{x}_k, t_n) - f_i^{*,eq}(\mathbf{x}_k, t_n)] \quad (4.9)$$

$$f_i^*(\mathbf{x}_k - \mathbf{c}_i \Delta t, t_{n-1}) = f_i^{*,col}(\mathbf{x}_k, t_n) \quad (4.10)$$

On remarque que les étapes de collision et de propagation sont très similaires à la LBM classique sauf que les particules se propagent en sens inverse ($-\mathbf{c}_i \Delta t$) et en temps inverse. L'emplacement du terme source dans le calcul adjoint correspond à l'endroit où la fonction coût est calculée dans le calcul direct. L'étape de collision (4.7) pour les nœuds appartenant à Γ possède donc un terme supplémentaire (qui est le terme source).

Il est aussi nécessaire d'enregistrer les champs directs \mathbf{u} et ρ afin de les relire en sens inverse au cours du calcul adjoint (pour le calcul du terme source dans l'adjoint et de la fonction d'équilibre adjointe), ce qui peut s'avérer trop lourd pour des cas industriels,

mais tout à fait réalisable pour ce cas test 2D.

La condition initiale pour le champ adjoint est choisie arbitrairement, d'après [94] :

$$f_i^*(\mathbf{x}, t_f) = 0 \quad (4.11)$$

La condition initiale du calcul adjoint est définie pour $t = t_f$, ce qui est bien cohérent puisque le calcul adjoint se fait de $t = t_f$ à $t = t_0$. Cette remarque prend toute son importance lors du calcul du gradient par l'adjoint (équation 2.48), par exemple $n=0$ correspond à la dernière valeur calculée de l'adjoint en réalité.

La fonction coût ne dépend pas explicitement de l'amplitude la source de contrôle (qui est le paramètre d'optimisation), donc nous avons $\frac{\partial I_0}{\partial \alpha} = 0$ dans l'équation (2.47). Le gradient peut s'écrire ainsi :

$$\nabla I_0 = \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \sum_{i=0}^{q-1} f_i^*(\mathbf{x}_k, t_{n+1}) \frac{\partial R_i(\mathbf{x}_k, t_n)}{\partial \alpha} \quad (4.12)$$

En considérant l'équation (4.1) pour calculer $\frac{\partial R_i(\mathbf{x}_k, t_n)}{\partial \alpha}$, la somme en espace disparaît car $\frac{\partial R_i(\mathbf{x}_k, t_n)}{\partial \alpha}$ est nul partout sauf sur le point source x_{kc} :

$$\nabla I_0 = \sum_{n=0}^{N-1} \sum_{i=0}^{q-1} f_i^*(\mathbf{x}_{kc}, t_{n+1}) \omega_i \sin(\Omega t_n) \quad (4.13)$$

4.1.2 Résultats

Champ direct

La figure 4.2 présente la densité à la fin du calcul (200ème pas de temps).

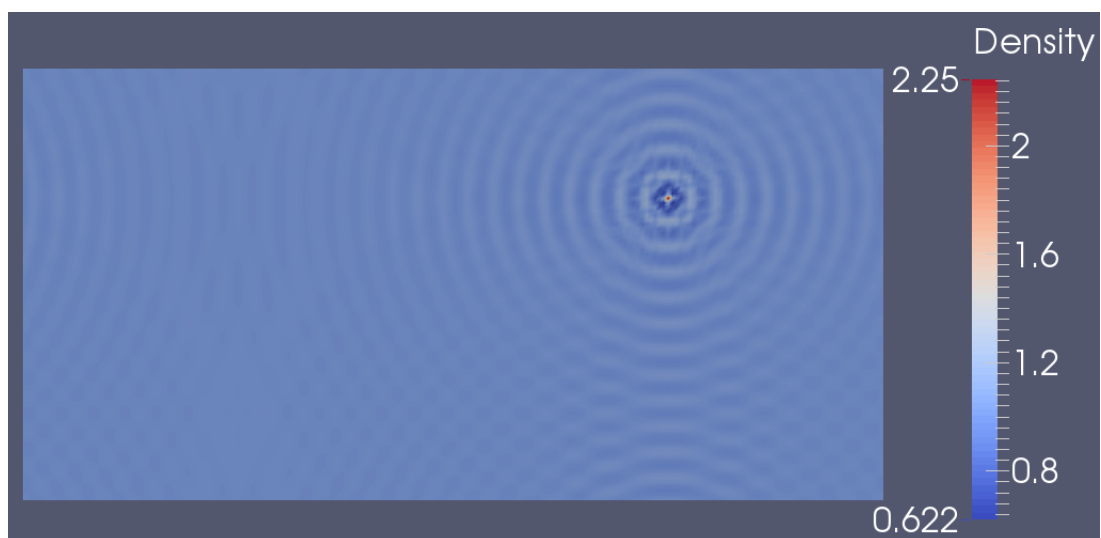


FIGURE 4.2 – densité au 200ème pas de temps

Champ adjoint

Les figures 4.3, 4.4, 4.5 et 4.6 présentent les densités adjointes pour respectivement les 25ème, 50ème, 75ème et 100ème pas de temps.

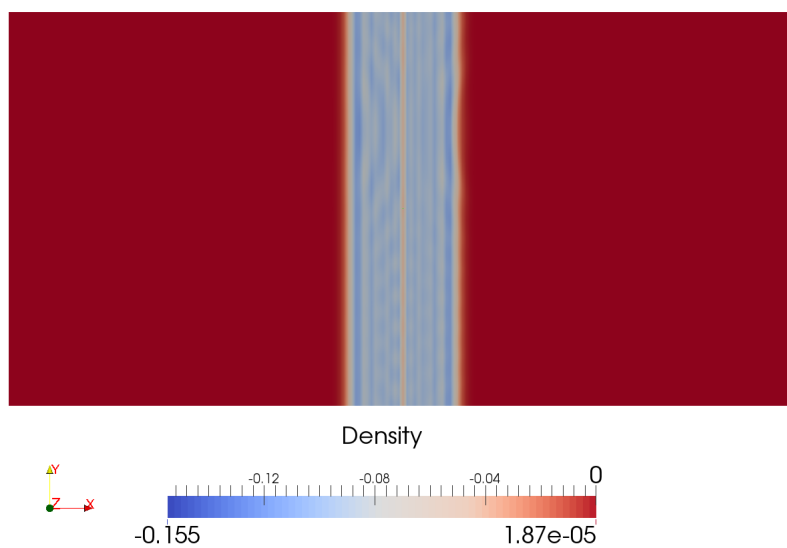


FIGURE 4.3 – densité adjointe au 25ème pas de temps

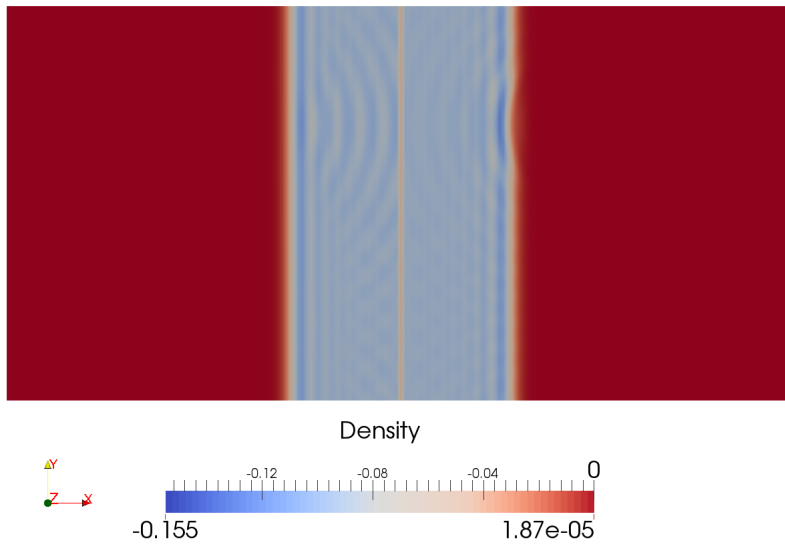


FIGURE 4.4 – densité adjointe au 50ème pas de temps

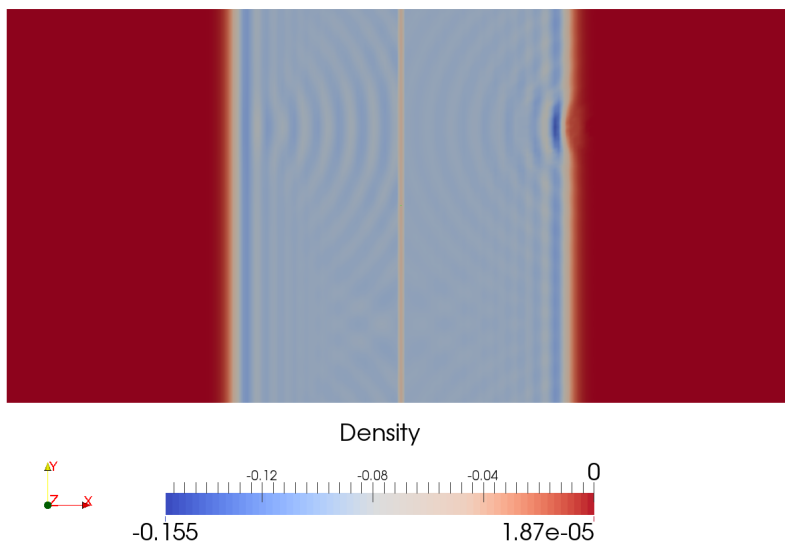


FIGURE 4.5 – densité adjointe au 75ème pas de temps

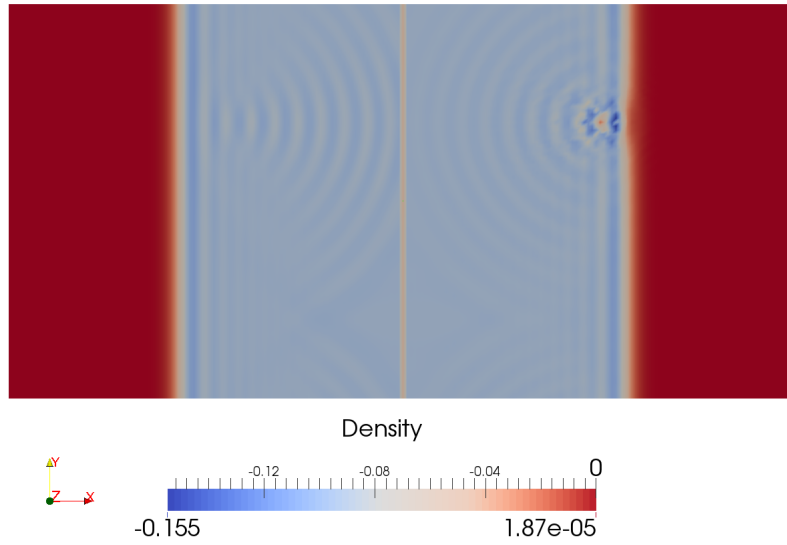


FIGURE 4.6 – densité adjointe au 100ème pas de temps

On peut remarquer dans les figures précédentes que la ligne Γ agit bien comme terme source dans le calcul adjoint, puisqu'on voit apparaître des ondes émanant de cette ligne dans le champ adjoint. On peut aussi remarquer que la relecture en temps inverse du champ direct est bien effectuée puisqu'on voit d'autres ondes émaner de l'emplacement de la source de bruit dans le champ adjoint.

Comparaison des gradients : différences finies et adjoint

Afin de vérifier la validité de la méthode présentée, on compare les gradients obtenus par adjoint à ceux obtenus par différences finies. Pour cela, on calcule le gradient par différences finies via :

$$\nabla I_0 = \frac{dI_0}{d\alpha} = \frac{I_0(\alpha + h) - I_0(\alpha)}{h} \quad (4.14)$$

avec α le paramètre d'optimisation qui varie entre 0 et 1 et h un petit pas fixé à 0,1. Il est important de remarquer qu'ici, le calcul du gradient par différences finies est simple car il n'y a qu'un seul paramètre d'optimisation.

La figure 4.7 présente l'évolution de I_0 en fonction de α , et on peut vérifier que la courbe du gradient (figure 4.8) est cohérente avec la courbe de la fonction coût associée.

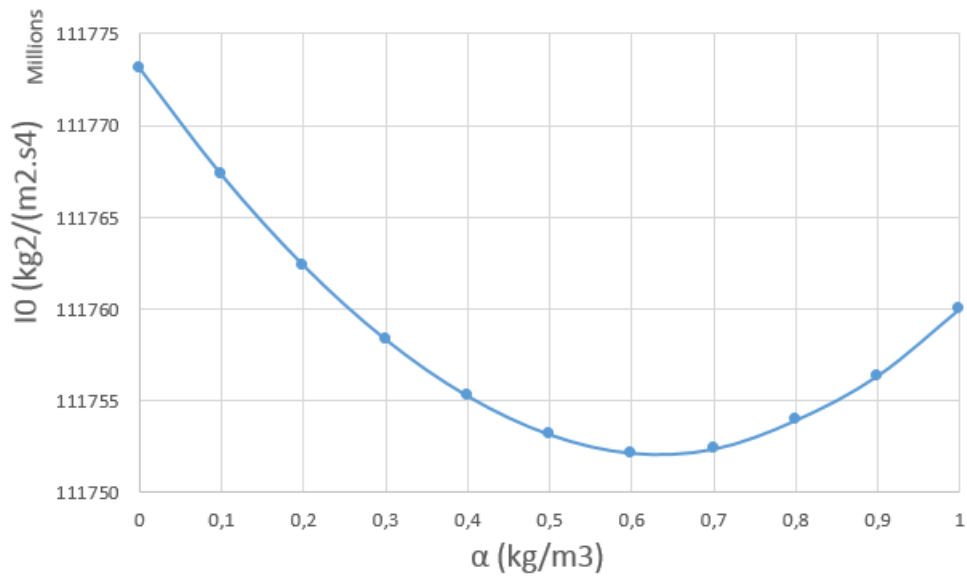


FIGURE 4.7 – Evolution de I_0 en fonction du paramètre d'optimisation

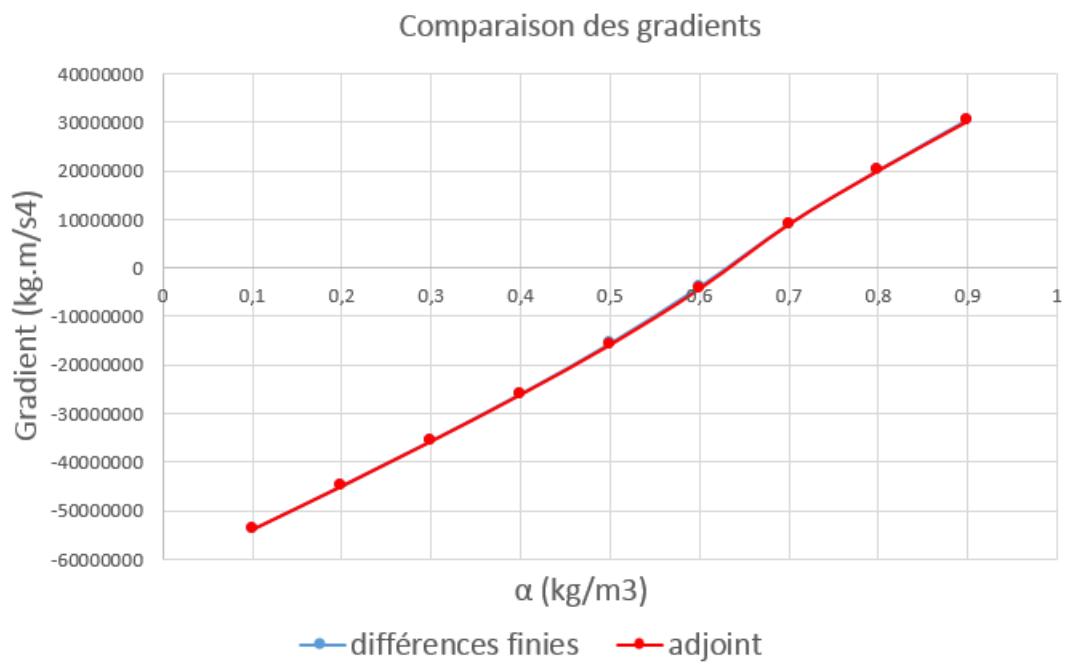


FIGURE 4.8 – Comparaison des gradients

L'inversion du signe du gradient pour $\alpha \in [0.5; 0.6]$ est cohérente avec le minimum de I_0 . On remarque que le gradient obtenu par l'adjoint est très proche de celui par

différences finies . Une fois que le gradient est obtenu, le paramètre α peut être modifié grâce à un algorithme de descente présenté dans la sous-section 1.2.3.

Ce premier cas nous a donc permis de valider l'implémentation du problème adjoint dans un cas simple, et de vérifier la validité des algorithmes codés pour créer le solveur adjoint.

4.2 Condition de rebond : Bounce-back

Nous allons à présent rajouter un obstacle dans le cas test précédent afin d'étudier les algorithmes de paroi implémentés à la surface de l'obstacle. Cela nous permettra de comparer différents algorithmes de paroi, et de valider les formulations adjointes de celles-ci. Le premier algorithme de paroi étudié est la condition bounce-back introduite dans 3.2.1.

4.2.1 Contrôle actif en acoustique en présence d'un obstacle : diminution du bruit

Problème direct

Le cas précédent est repris ici, en ajoutant un obstacle comme le montre la figure 4.9, ce qui permet de vérifier que l'algorithme de paroi adjoint est correct. L'obstacle est un cylindre de rayon 0.2 m à la position (2,5 ; 1,4). Une difficulté supplémentaire apparaît alors, à savoir l'implémentation d'une condition limite sur l'obstacle pour le calcul adjoint. Le reste de la mise en données est identique : l'objectif est toujours de réduire le bruit sur la ligne Γ , en ayant comme paramètre d'optimisation l'amplitude de la source de contrôle.

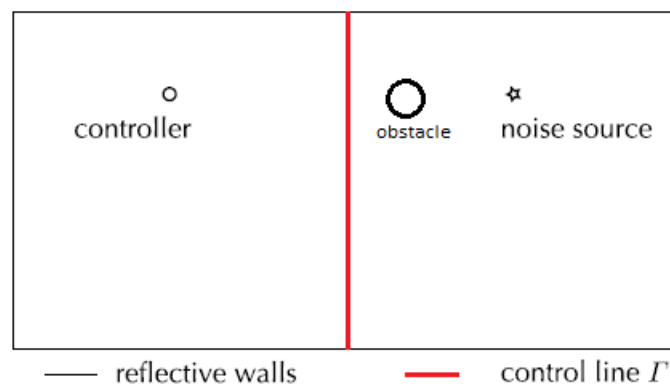


FIGURE 4.9 – Schéma du problème acoustique

Sachant que la mise en données est identique au cas précédent, seule l'implémentation de la condition limite sur l'obstacle sera détaillée par la suite, ainsi que sa formulation

adjointe.

Problème adjoint : Développement du link bounce-back adjoint

Au niveau de l'obstacle, la condition de link bounce-back a été choisie dans un premier temps, car elle est relativement simple à implémenter. L'obtention de sa formulation adjointe est décrite ici. Nous reprenons la démarche utilisée dans la sous-section 2.2.4 : nous utilisons la forme discrétisée des équations d'état (c'est-à-dire nous remplaçons R par $R_j(\mathbf{x}_k, t_n)$), puis nous dérivons par rapport aux variables d'état ($f_i(\mathbf{x}_{k0}, t_{n0})$) en se plaçant à un nœud spécifique du maillage $\mathbf{x} = \mathbf{x}_{k0}$ (ici, \mathbf{x}_{k0} est nécessairement un nœud proche paroi puisque l'on s'intéresse à un algorithme utilisé proche paroi) et à un temps donné $t = t_{n0}$. Cela permet de simplifier le calcul puisqu'il ne reste quasiment que des Dirac lorsqu'on dérive.

Tout d'abord, les résidus de la LBM s'écrivent à la paroi en utilisant le bounce-back classique :

$$R_j(\mathbf{x}_k, t_n) = f_j(\mathbf{x}_k, t_{n+1}) - f_{opp(j)}^{col}(\mathbf{x}_k, t_n) = 0 \quad (4.15)$$

avec la fonction de collision définie de la façon suivante :

$$f_{opp(j)}^{col}(\mathbf{x}, t) = f_{opp(j)}(\mathbf{x}_k, t_n) - \frac{1}{\tau} [f_{opp(j)}(\mathbf{x}_k, t_n) - f_{opp(j)}^{eq}(\mathbf{x}_k, t_n)] \quad (4.16)$$

Lorsqu'on dérive par rapport à $f_i(\mathbf{x}_{k0}, t_{n0})$ on obtient :

$$\frac{\partial R_j(\mathbf{x}_k, t_n)}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = \delta_{j,i} \delta_{\mathbf{x}_k, \mathbf{x}_{k0}} \delta_{n_0, n+1} - \left(1 - \frac{1}{\tau}\right) \delta_{opp(j), i} \delta_{\mathbf{x}_k, \mathbf{x}_{k0}} \delta_{n_0, n} - \frac{1}{\tau} \frac{\partial f_{opp(j)}^{eq}}{\partial f_i}(\mathbf{x}_{k0}, t_{n0}) = 0 \quad (4.17)$$

Sachant que I_0 n'est pas défini sur l'obstacle, il n'y a pas de dépendance explicite de I_0 à $f_i(\mathbf{x}_{k0}, t_{n0})$ et $\frac{\partial I_0}{\partial f_i} = 0$. En utilisant l'équation 2.34, les équations adjointes proche paroi s'écrivent alors :

$$\sum_j \sum_k \sum_n f_j^*(\mathbf{x}_k, t_{n+1}) \frac{\partial R_j(\mathbf{x}_k, t_n)}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = 0 \quad (4.18)$$

Nous pouvons alors écrire :

$$f_i^*(\mathbf{x}_{k0}, t_{n0}) - \left(1 - \frac{1}{\tau}\right) f_{opp(i)}^*(\mathbf{x}_{k0}, t_{n0+1}) - \frac{1}{\tau} \sum_j f_j^*(\mathbf{x}_{k0}, t_{n0+1}) \frac{\partial f_{opp(j)}^{eq}}{\partial f_i}(\mathbf{x}_{k0}, t_{n0}) = 0 \quad (4.19)$$

Nous cherchons à présent à calculer le terme $\sum_j f_j^*(\mathbf{x}_{k0}, t_{n0+1}) \frac{\partial f_{opp(j)}^{eq}}{\partial f_i}(\mathbf{x}_{k0}, t_{n0})$ que l'on appelle fonction d'équilibre adjointe pour le link bounce-back et notée $f_{opp(i), LBB}^{*, eq}$.

Sachant que $\omega_{opp(j)} = \omega_j$ et que $\mathbf{c}_{opp(j)} = -\mathbf{c}_j$, on obtient :

$$f_{opp(j)}^{eq} = \rho\omega_j \left(1 - \frac{\mathbf{c}_j \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_j \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right) \quad (4.20)$$

$$\frac{\partial f_{opp(j)}^{eq}}{\partial f_i} = \omega_i \left(1 - \frac{\mathbf{c}_i \cdot \mathbf{c}_j}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{c}_i \mathbf{u}) \right) \quad (4.21)$$

En reprenant le raisonnement utilisé dans la sous-section 2.2.4 :

$$f_{opp(i),LBB}^{*,eq} = \sum_{j=1}^{19} \omega_j f_j^* \left[\left(1 - \frac{\mathbf{c}_j \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{u} \otimes \mathbf{u}) \right) + \left(-\mathbf{c}_j + \frac{1}{c_s^2} \mathcal{H}_j^{(2)} \right) \cdot \frac{\mathbf{c}_i - \mathbf{u}}{c_s^2} \right] \quad (4.22)$$

La fonction d'équilibre adjointe pour la condition limite link bounce-back peut ainsi s'écrire de la façon suivante :

$$f_{i,LBB}^{*,eq} = \rho_{LBB}^* + \frac{(-\mathbf{c}_i - \mathbf{u}) \cdot \mathbf{j}_{LBB}^*}{c_s^2} \quad (4.23)$$

en définissant les variables adjointes suivantes :

$$\rho_{LBB}^* = \sum_{j=1}^{19} \omega_j f_j^* \left(1 - \frac{\mathbf{c}_j \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_j^{(2)} : (\mathbf{u} \otimes \mathbf{u}) \right) \quad (4.24)$$

$$\mathbf{j}_{LBB}^* = \sum_{j=1}^{19} \omega_j f_j^* \left(-\mathbf{c}_j + \frac{1}{c_s^2} \mathcal{H}_j^{(2)} \mathbf{u} \right) \quad (4.25)$$

La condition de link bounce-back au niveau de l'obstacle s'écrit alors pour le problème adjoint :

$$f_j^*(\mathbf{x}_k, t_n) = f_{opp(j)}^{*,col,LBB}(\mathbf{x}_k, t_{n+1}) \quad (4.26)$$

avec

$$f_{opp(j)}^{*,col,LBB} = \left(1 - \frac{1}{\tau} \right) f_{opp(j)}^* + \frac{1}{\tau} f_{opp(j),LBB}^{*,eq} \quad (4.27)$$

On remarque que ces équations sont très similaires au link bounce-back standard, à l'exception du signe - dans les équations 4.23, 4.24 et 4.25 par rapport aux équations 2.44, 2.45 et 2.46.

4.2.2 Résultats

Les résultats présentés dans cette section concernent l'utilisation du link bounce-back au niveau de la paroi dans le calcul direct, ainsi que l'utilisation du link bounce-back adjoint dans le calcul adjoint.

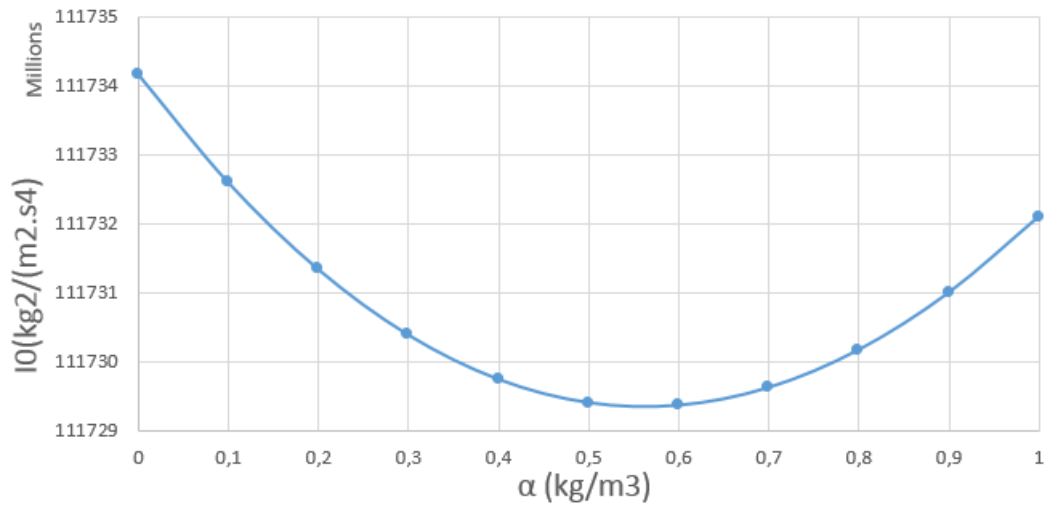


FIGURE 4.10 – Evolution de I_0 en fonction du paramètre d'optimisation

On remarque que la fonction coût étudiée est minimale pour $\alpha \in [0.5; 0.6]$, ce qui est cohérent avec le fait que le gradient de la fonction coût s'annule pour $\alpha \in [0.5; 0.6]$ aussi :

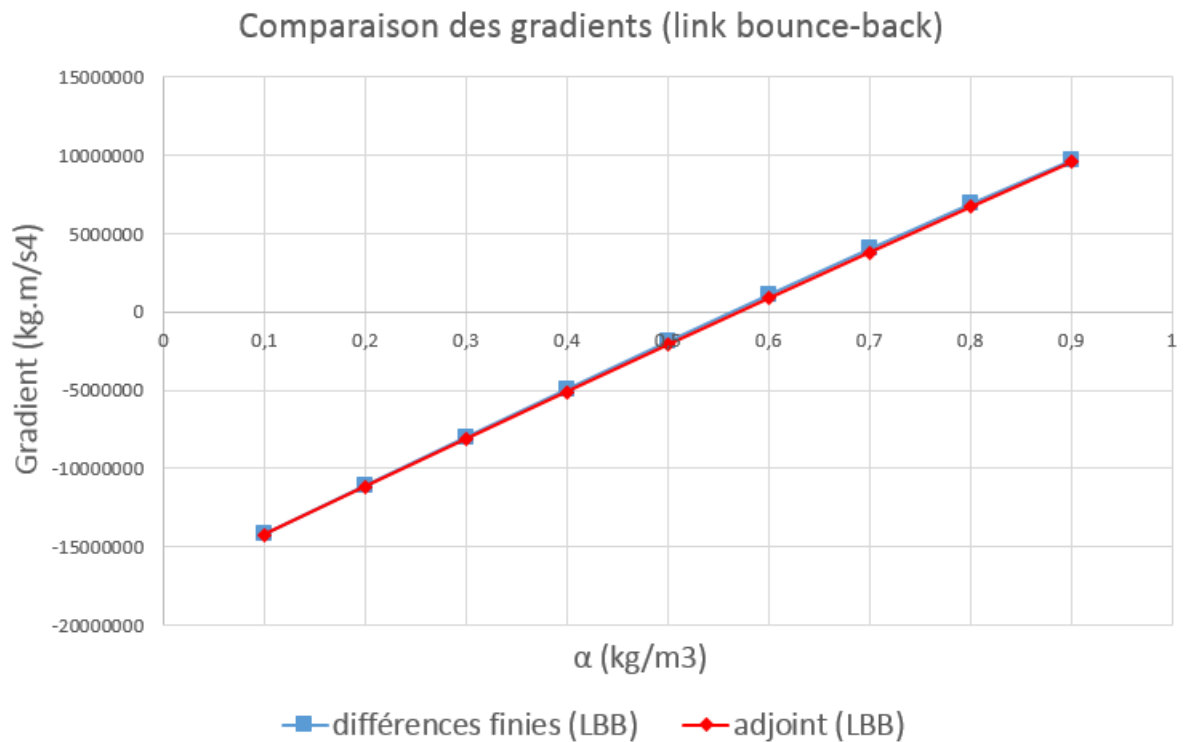


FIGURE 4.11 – Comparaison des gradients

On remarque que les courbes des gradients sont cohérentes avec la courbe de la fonction coût. Le gradient par adjoint est très proche de celui par différences finies, ce qui permet de valider l'implémentation du link bounce-back adjoint.

4.3 Méthode de reconstruction utilisée dans ProLB

L'algorithme de paroi qui est utilisé dans le logiciel ProLB est étudié dans cette partie, à savoir l'algorithme de paroi qui a été présenté dans 3.2.3. Cependant, nous allons proposer une simplification concernant l'algorithme de paroi adjoint, afin de ne pas avoir à dériver l'algorithme de paroi de ProLB qui semble complexe à dériver.

4.3.1 Contrôle actif en acoustique en présence d'un obstacle : diminution du bruit

Pour tester l'algorithme de paroi de ProLB, nous reprenons la mise en données du cas précédent. Ce cas test nous permettra de valider l'hypothèse suivante : des gradients corrects peuvent être obtenus même si l'algorithme de paroi adjoint n'est pas dérivé de l'algorithme de paroi direct. En effet, l'algorithme de paroi de ProLB décrit dans la sous-section 3.2.3 étant trop complexe pour être dérivée afin d'obtenir son équivalent adjoint, nous avons décidé d'utiliser un algorithme de paroi différent pour le calcul adjoint. Cela simplifie grandement la démarche, et nous verrons si cette hypothèse est utilisable pour le cas académique ici et plus tard dans le cas industriel.

Problème direct

La mise en données est identique à celle du cas précédent, à l'exception de l'algorithme de paroi dans le problème direct qui est celui expliqué dans la partie bibliographique sur ProLB (3.2.3).

Problème adjoint

Concernant la mise en données du problème d'optimisation, on reprend celle utilisée précédemment : l'objectif est toujours de réduire le bruit sur la ligne Γ , en ayant comme paramètre d'optimisation l'amplitude de la source de contrôle. Cependant ici, l'algorithme de paroi adjoint n'est pas dérivé de l'algorithme de paroi direct. Dans ce cas, on utilise l'algorithme de paroi adjoint du link bounce-back pour l'adjoint (équation 4.26).

4.3.2 Résultats

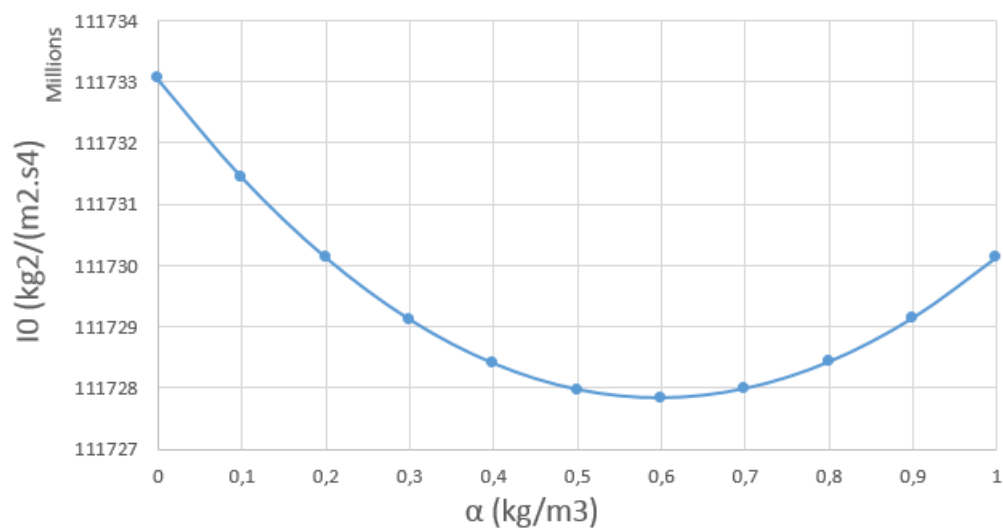


FIGURE 4.12 – Evolution de I_0 en fonction du paramètre d'optimisation

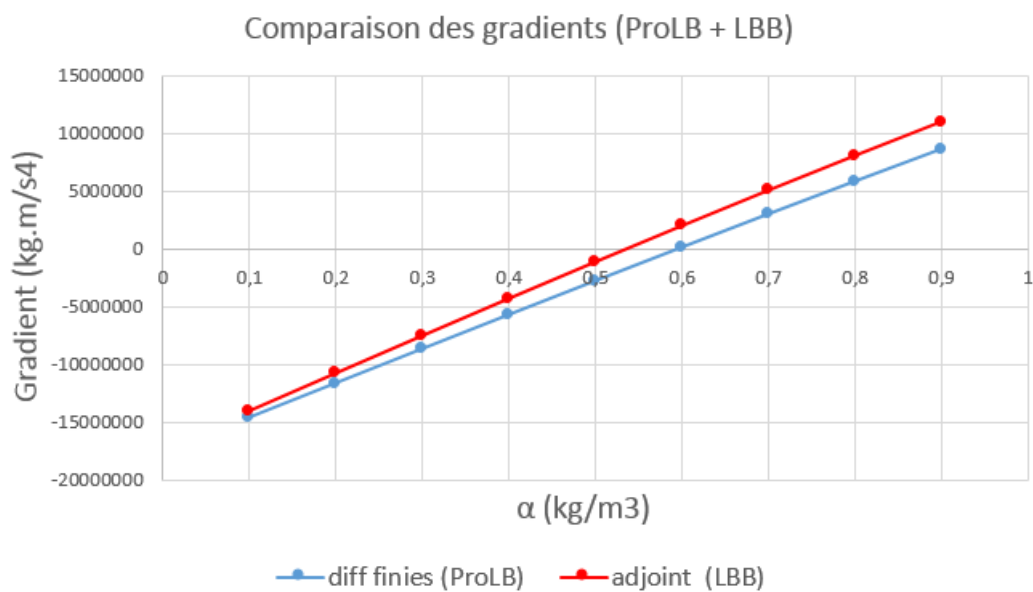


FIGURE 4.13 – Comparaison des gradients par différences finies et adjoint

On remarque ici que les écarts entre les gradients par différences finies et par adjoint sont plus élevés que dans les cas précédents, mais l'écart reste acceptable pour une utilisation industrielle. De plus, ne pas avoir à différencier l'algorithme de paroi implémenté

dans ProLB simplifie grandement la démarche et permet d'envisager une application industrielle donnant de bons résultats.

4.4 Interpolation de Ginzburg

Cette partie nous permet de présenter à la fois un nouveau cas d'étude, et un nouvel algorithme de paroi. L'algorithme de paroi de Ginzburg présenté dans 3.2.2 est appliqué au même obstacle que dans le cas précédent, dans la même géométrie, à l'exception du fait que la fonction coût change ici comme nous le verrons par la suite. L'introduction de ce nouvel algorithme de paroi vient du fait que le "bounce-back" est attractif pour sa simplicité, mais il ne fait pas intervenir la distance à la paroi ce qui rend le calcul du gradient nul dans le cas où on calcule la force de traînée dans le sillage (puisque $\frac{\partial I_0}{\partial \alpha} = 0$ et $\frac{\partial R}{\partial \alpha} = 0$).

4.4.1 Contrôle actif en acoustique en présence d'un obstacle : diminution des efforts sur l'obstacle

La géométrie du cas précédent est reprise ici, mais on souhaite désormais optimiser les efforts de pression sur l'obstacle. Ce cas se rapproche donc des applications automobiles pour lesquelles on souhaite minimiser la force de traînée automobile sur le véhicule. La fonction coût est alors calculée sur l'obstacle comme le montre le schéma suivant :

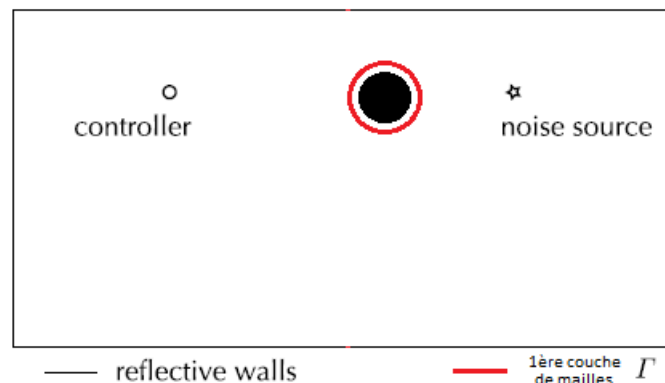


FIGURE 4.14 – Schéma du problème acoustique

En réalité, les efforts de pression sont calculés sur la ligne rouge entourant l'obstacle, qui représente la première couche d'éléments. Cette méthode est adoptée car elle évite d'interpoler les valeurs de pression sur l'obstacle comme nous le verrons par la suite. Pour procéder par étapes, le paramètre d'optimisation est toujours l'amplitude de la source de contrôle, ce qui permet de comparer simplement le gradient par différences finies à celui par l'adjoint. Par la suite, un paramètre d'optimisation relatif à la forme de l'obstacle sera pris en compte, ce qui correspond plus à nos applications automobiles

pour lesquelles nous souhaitons modifier la forme de l'obstacle pour réduire la traînée aérodynamique.

Problème direct

Les efforts de pression sur un obstacle peuvent être calculés en sommant les pressions projetées sur chaque élément de l'obstacle (on néglige les efforts visqueux dans un premier temps) :

$$I_0 = - \sum_f (p_f \vec{n}_f \cdot \vec{x} a_f) \quad (4.28)$$

avec f les faces appartenant à l'obstacle, p_f la pression interpolée sur la face, \vec{n}_f le vecteur normal à la face et orienté vers l'extérieur de l'obstacle, a_f l'aire de la face, et \vec{x} l'axe horizontal puisque l'on souhaite minimiser les efforts de pression selon cet axe. Dans notre approche, nous avons fait le choix d'adopter une méthode différente puisque l'on calcule cette force sur la première couche de maille entourant l'obstacle et non pas sur l'obstacle directement. Cela permet de ne pas avoir à interpoler les valeurs de pression sur chaque élément de l'obstacle. Cela impose cependant de compter le nombre de nœuds qui se projettent sur un même élément de l'obstacle, afin de calculer une force équivalente. Pour cela, un algorithme a été développé permettant de compter le nombre de nœuds (en rouge) associés à chaque face de l'obstacle, comme le montre le schéma suivant :

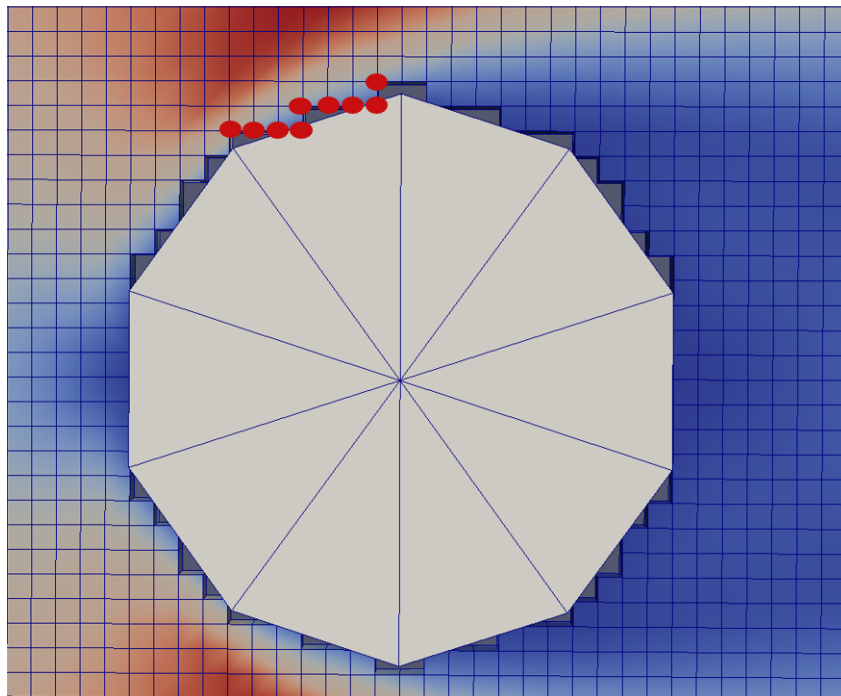


FIGURE 4.15 – Schéma du problème acoustique

On voit par exemple ici que 9 nœuds ont une projection sur la même face, il faudra donc diviser l'aire de cette face par 9 lorsque l'on fait la somme sur la première couche de mailles. On notera $\overline{a_f}$ l'aire de la face pondérée par le nombre de nœuds qui s'y appliquent, par exemple dans le cas expliqué : $\overline{a_f} = \frac{a_f}{9}$

Dans cette étude, on choisit de minimiser les efforts de pression sommés à la fois en espace et en temps, pour se rapprocher de la fonction coût utilisée précédemment. La nouvelle fonction coût utilisée pour ce problème est alors :

$$I_0 = - \sum_k \sum_n (p_k \vec{n}_f \cdot \vec{x} \overline{a_f}) \quad (4.29)$$

avec $k \in \Gamma$ (première couche de maillage entourant l'obstacle) et $n \in T$.

Problème adjoint : Développement de l'interpolation de Ginzburg adjoint

On rappelle que l'interpolation de Ginzburg [30] s'écrit :

$$f_j(\mathbf{x}_k, t_{n+1}) = f_{opp(j)}^{col}(\mathbf{x}_k, t_n) + \frac{1-2d}{1+2d} [f_{opp(j)}^{col}(\mathbf{x}_k + \mathbf{c}_j \Delta t, t_n) - f_j^{col}(\mathbf{x}_k, t_n)] \quad (4.30)$$

En reprenant un développement totalement similaire à celui fait pour l'établissement de l'algorithme de LBB adjoint (section 4.2), on obtient l'algorithme de Ginzburg adjoint suivant :

$$f_j^*(\mathbf{x}_k, t_n) = f_{opp(j)}^{*col,LBB}(\mathbf{x}_k, t_{n+1}) + \frac{1-2d}{1+2d} [f_{opp(j)}^{*col,LBB}(\mathbf{x}_k - \mathbf{c}_j \Delta t, t_{n+1}) - f_j^{*col}(\mathbf{x}_k, t_{n+1})] \quad (4.31)$$

avec $f_{opp(j)}^{*col,LBB}$ la fonction de collision spéciale pour le link bounce-back calculée dans la section précédente (équation 4.27).

Le terme source à implémenter dans le calcul adjoint est différent des cas précédents puisque nous étudions une fonction coût différente. Il faut calculer $\frac{\partial I_0}{\partial f}$ pour les nœuds appartenant à Γ . Sachant que l'on peut écrire $p = \rho c_s^2$, et $\rho = \sum_j f_j$, on obtient pour $x_k \in \Gamma$ et $t_n \in T$:

$$\frac{\partial I_0}{\partial f_i} = - \frac{\partial (c_s^2 (\sum_j f_j) \vec{n}_f \cdot \vec{x} \overline{a_f})}{\partial f_i} = -c_s^2 N_x \overline{a_f} \quad (4.32)$$

avec $N_x = \vec{n}_f \cdot \vec{x}$.

Ce terme est implémenté dans le calcul adjoint comme un terme source pour les nœuds appartenant à Γ . Deux équations différentes peuvent alors s'appliquer selon que la direction de propagation est "solide" ou "fluide". Une direction "solide" est une direction pour laquelle le nœud voisin est solide. De la même façon, une direction "fluide" est une

direction pour laquelle le nœud voisin est fluide. Pour une direction fluide, l'équation proche paroi devient :

$$f_i^*(\mathbf{x} - \mathbf{c}_i \Delta t, t - \Delta t) = f_i^*(\mathbf{x}, t) - \frac{1}{\tau} [f_i^*(\mathbf{x}, t) - f_i^{*,eq}(\mathbf{x}, t)] + c_s^2 N_x \bar{a}_f \quad (4.33)$$

Pour une direction solide, sachant que l'on utilise le modèle adjoint de l'interpolation de Ginzburg (4.31) à la paroi, nous avons :

$$f_j^*(\mathbf{x}_k, t_n) = f_{opp(j)}^{*col,LBB}(\mathbf{x}_k, t_{n+1}) + \frac{1-2d}{1+2d} [f_{opp(j)}^{*col,LBB}(\mathbf{x}_k - \mathbf{c}_j \Delta t, t_{n+1}) - f_j^{*col}(\mathbf{x}_k, t_{n+1})] + c_s^2 N_x \bar{a}_f \quad (4.34)$$

4.4.2 Résultats

Puisque la fonction coût est définie sur l'obstacle, le terme source dans l'adjoint apparaît aussi sur l'obstacle ce qui est visible sur les champs de vitesse et densité adjointes :

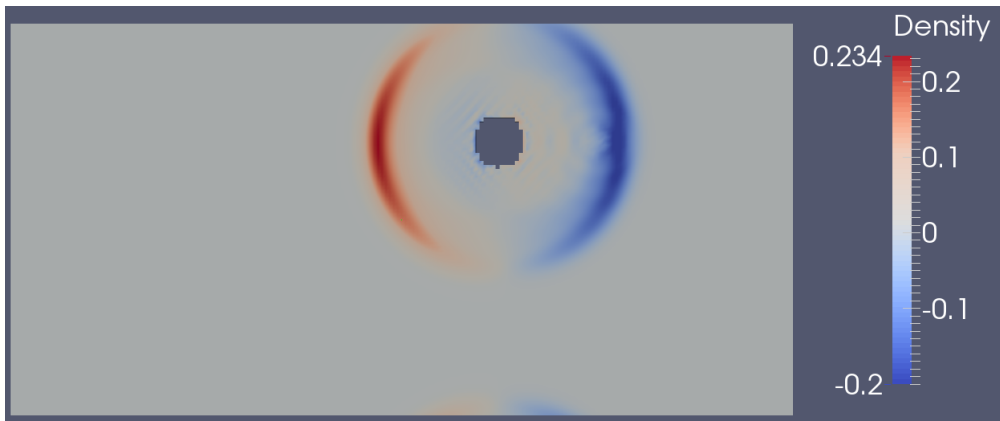


FIGURE 4.16 – densité adjointe à la 50° it

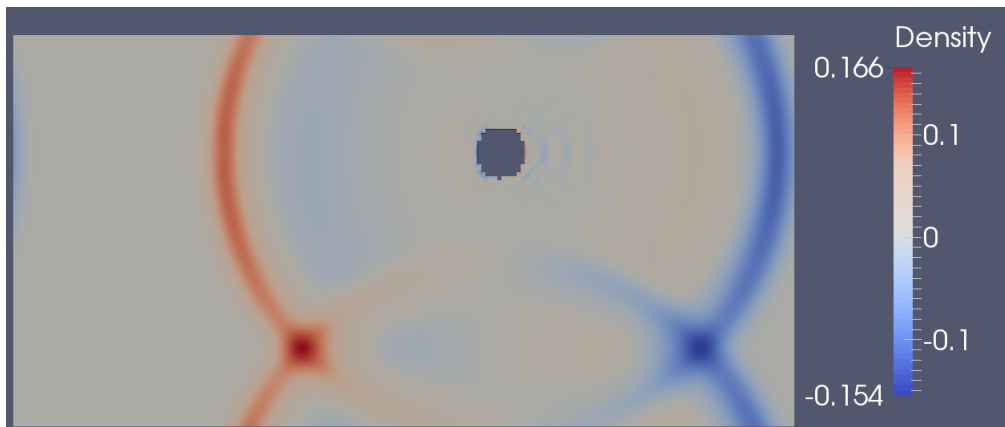


FIGURE 4.17 – densité adjointe à la 200° it

La courbe de la fonction coût en fonction du paramètre d'optimisation est la suivante :

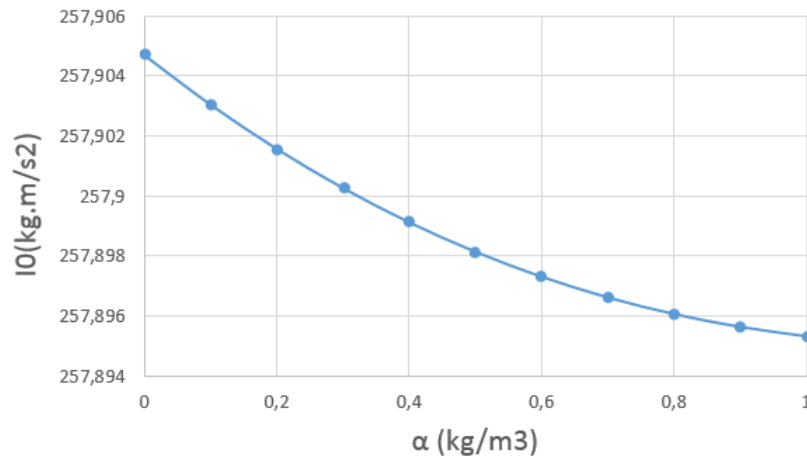


FIGURE 4.18 – Evolution de I_0 en fonction du paramètre d'optimisation

On remarque que la courbe de I_0 ne présente pas d'extremum dans cette plage de valeurs de α , mais l'intérêt de ce cas test étant de valider l'implémentation du calcul des forces de pression, cela a peu d'importance. Il convient cependant de vérifier que la courbe du gradient est cohérente avec celle-ci, en vérifiant que le gradient ne s'annule pas.

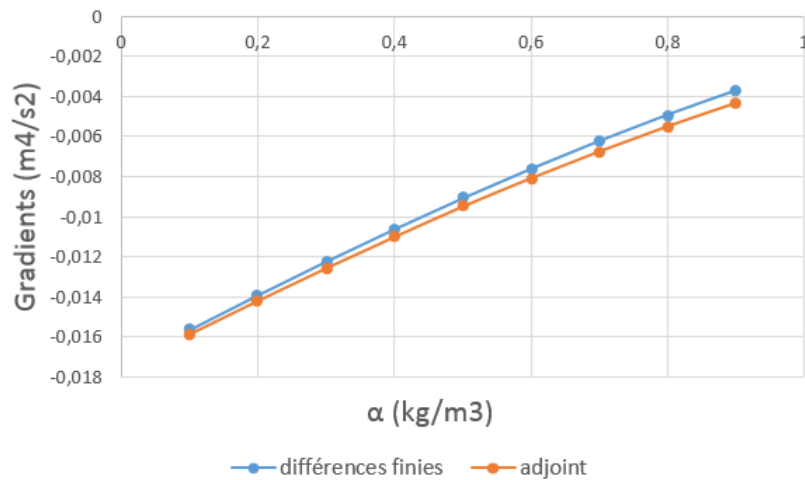


FIGURE 4.19 – Comparaison des gradients

Le gradient reste négatif pour cette plage de valeurs de α , et est cohérente avec la courbe de I_0 . De plus, la courbe du gradient par adjoint est très proche de celle du gradient par différences finies, ce qui permet de valider le développement de l'algorithme de paroi de Ginzburg pour l'adjoint.

4.5 Raffinements de maillage

L'utilisation de raffinements de maillage est indispensable pour des cas industriels pour lesquels utiliser un maillage uniforme avec une seule taille de maille mènerait à des calculs trop onéreux à résoudre. Cela implique d'utiliser des algorithmes spécifiques traitant les fonctions de distribution traversant une interface entre maillage grossier et fin (respectivement les points rouges et bleus sur la figure 3.3).

4.5.1 Algorithmes du problème direct

Pour rappel, la première chose à prendre en compte est le fait que les variables macroscopiques, telles que la vitesse et la densité, sont continues sur tout le domaine de calcul et plus spécifiquement sur les interfaces entre maillage grossier et fin. D'un autre côté, les fonctions de distributions sont discontinues sur ces mêmes interfaces, et doivent donc être recalées quand elles traversent ces interfaces. Les équations suivantes, inspirées de [23], [22] et [50], sont utilisées ici, avec f_i^c la fonction sur le nœud grossier et f_i^f celle sur le nœud fin :

$$f_i^f(\mathbf{x}_k, t_n) = \frac{1}{2\tau_f + 1} ((2\tau_f)f_i^c(\mathbf{x}_k, t_n) + f_i^{eq}(\mathbf{x}_k, t_n)) \quad (4.35)$$

$$f_i^c(\mathbf{x}_k, t_n) = \frac{1}{2\tau_f} ((2\tau_f + 1)f_i^f(\mathbf{x}_k, t_n) - f_i^{eq}(\mathbf{x}_k, t_n)) \quad (4.36)$$

Pour utiliser ces équations, il faut connaître les fonctions d'équilibre sur les nœuds en question. Sachant que les variables macroscopiques sont indépendantes de la taille de maille, les fonctions d'équilibre n'en dépendent pas non plus. Nous avons donc sur une interface : $f_{eq}^f = f_{eq}^c = f_{eq}$. Les fonctions de recalage nécessitent aussi les temps de relaxation locaux. La viscosité étant continue à l'interface, le temps de relaxation doit être modifié de façon adéquate :

$$\tau^f = 2\tau^c - \frac{1}{2} \quad (4.37)$$

Nous rappelons aussi que dans la LBM, la relation $\frac{dx}{dt}$ (pas d'espace / pas de temps) doit être constante et que l'interpolation temporelle suivante est utilisée pour calculer les valeurs des fonctions de distribution sur les nœuds fins manquants :

$$f_i^f(\mathbf{x}_k, t_n) = \alpha f_i^f(\mathbf{x}_k, t_{n-2}) + \beta f_i^f(\mathbf{x}_k, t_{n-1}) + \gamma f_i^f(\mathbf{x}_k, t_{n+1}) \quad (4.38)$$

Les coefficients α , β et γ sont fixés ici.

Enfin, les fonctions allant du maillage grossier vers fin sur les nœuds qui ne coïncident pas sur l'interface sont inconnues (figure 4.20). Pour cela on va utiliser une interpolation d'ordre 4 centrée (voir figure 4.20), ou bien, proche des parois ou des limites du domaine, on réduit l'interpolation avec un schéma sur 3 points décentrés, ou bien un schéma centré sur 2 points.

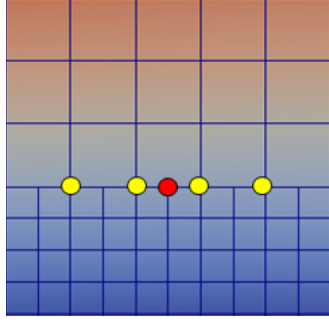


FIGURE 4.20 – nœuds utilisés pour l’interpolation spatiale d’ordre 4

Une particularité à noter ici est que ces interpolations, spatiales et temporelles, sont effectués sur les variables macroscopique (vitesse et densité) dans le solveur ProLB standard. Ici, il a été décidé d’utiliser d’interpoler les fonctions de distributions et non les variables macroscopiques, car cela facilite la dérivation des modèles pour obtenir leur formulation adjointe.

4.5.2 Algorithmes du problème adjoint

Nous reprenons exactement la même démarche que dans la sous-section 2.2.4. La première étape consiste à écrire les équations des transitions de maillage sous forme résiduelle. Si nous prenons par exemple le rescaling des fonctions de la grille grossière vers fine :

$$R_j(\mathbf{x}_k, t_n) = f_j^f(\mathbf{x}_k, t_n) - \frac{1}{2\tau_f + 1}((2\tau_f)f_j^c(\mathbf{x}_k, t_n) + f_j^{eq}(\mathbf{x}_k, t_n)) = 0 \quad (4.39)$$

Ensuite, il faut dériver $R_j(\mathbf{x}_k, t_n)$ par rapport à $f_i(\mathbf{x}_{k0}, t_{n0})$:

$$\frac{\partial R_j(\mathbf{x}_k, t_n)}{\partial f_i(\mathbf{x}_{k0}, t_{n0})} = \delta_{ji}\delta_{x_k, x_{k0}}\delta_{n_0, n} - \frac{1}{2\tau_f + 1}((2\tau_f)\delta_{ji}\delta_{x_k, x_{k0}}\delta_{n_0, n} + \frac{\partial f_j^{eq}}{\partial f_i}\delta_{x_k, x_{k0}}\delta_{n_0, n_n}) \quad (4.40)$$

Il convient maintenant de multiplier l’équation précédente par f_j^* afin de trouver l’équation adjointe puis de sommer en espace, temps et vitesse. Nous avons déjà calculé le terme $\sum_j f_j^*(x_{k0}, t_{n0+1})\frac{\partial f_j^{eq}}{\partial f_i}(x_{k0}, t_{n0})$ que nous avons appelé $f_i^{*,eq}(x_{k0}, t_{n0+1})$ dans la section 2.2.4. Sachant que la fonction coût est calculée en dehors des zones de transition, il n’y a pas de terme source au niveau des transitions et nous trouvons ainsi le rescaling adjoint pour les fonctions adjointes de la grille grossière vers fine :

$$f_i^{*f}(\mathbf{x}_k, t_n) = \frac{1}{2\tau_f + 1}((2\tau_f)f_i^{*c}(\mathbf{x}_k, t_n) + f_i^{*,eq}(\mathbf{x}_k, t_n)) \quad (4.41)$$

La même démarche mène au rescaling adjoint des fonctions adjointes de la grille fine vers grossière :

$$f_i^{*c}(\mathbf{x}_k, t_n) = \frac{1}{2\tau_f} ((2\tau_f + 1)f_i^{*f}(\mathbf{x}_k, t_n) - f_i^{*eq}(\mathbf{x}_k, t_n)) \quad (4.42)$$

Enfin, la même démarche de calcul du résidu conduit également à l'interpolation temporelle adjoint, qui est identique à celle directe, à l'exception que nous manipulons ici des fonctions de distributions adjointes :

$$f_i^{*f}(\mathbf{x}_k, t_n) = \alpha f_i^{*f}(\mathbf{x}_k, t_{n+2}) + \beta f_i^{*f}(\mathbf{x}_k, t_{n+1}) + \gamma f_i^{*f}(\mathbf{x}_k, t_{n-1}) \quad (4.43)$$

De façon similaire, les interpolations spatiales adjointes pour les nœuds fins sur une interface sont identiques à celles utilisées pour le problème direct.

4.5.3 Résultats

Pour vérifier la validité des algorithmes utilisés pour les raffinements de maillage dans le problème direct mais aussi adjoint, nous étudions l'écoulement autour d'un cylindre en 2D. Le maillage contient quatre zones de résolution différentes, chacune englobant le cylindre. Cette section permet seulement de valider les algorithmes implementés dans le calcul direct et adjoint au niveau des raffinements de maillage car la boucle complète d'optimisation sera étudiée dans la section 5.2. La mise en données est la même que celle de la section 5.2.

La figure 4.21 montre le champ de vitesse derrière le cylindre.

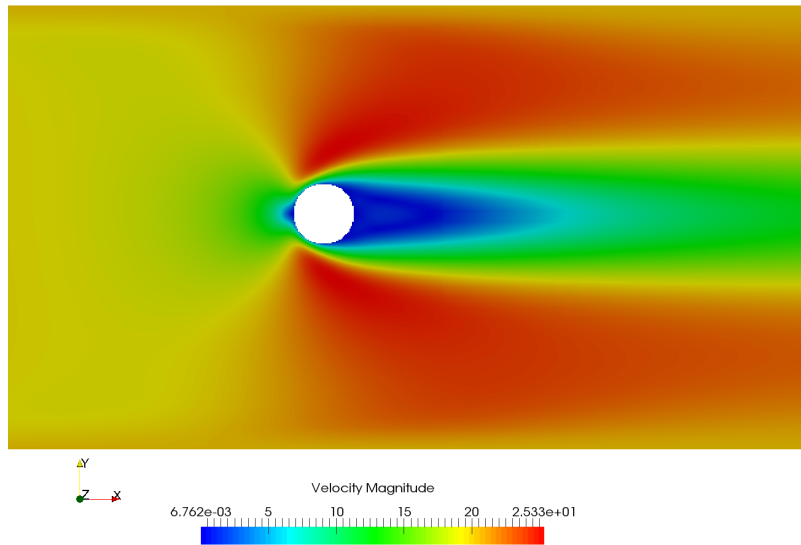


FIGURE 4.21 – Champ de vitesse direct

Nous pouvons remarquer que l'écoulement est continu et que les algorithmes utilisés semblent prédire correctement le comportement laminaire du fluide derrière le cylindre. Nous pouvons aussi visualiser le maillage utilisé pour ce calcul :

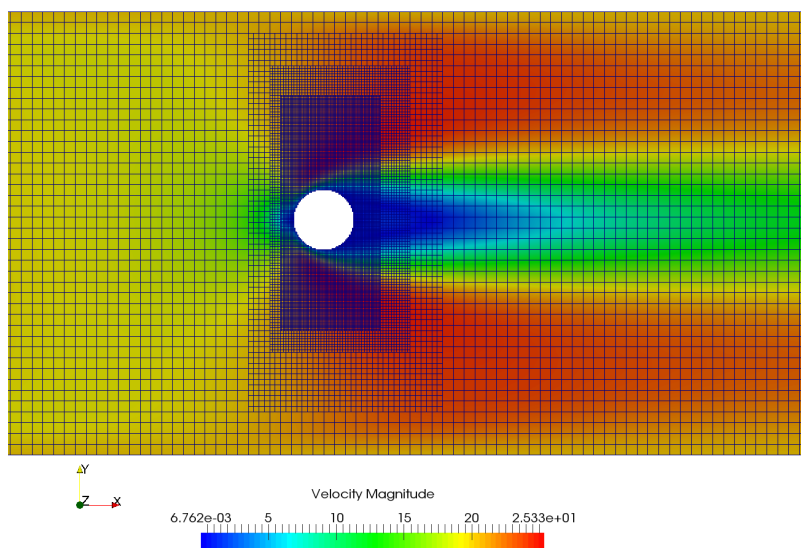


FIGURE 4.22 – Superposition du maillage et du champ de vitesse direct

La figure 4.22 nous montre que les algorithmes utilisés pour les raffinements de maillage dans le problème direct sont corrects et ne donnent lieu à aucune discontinuité.

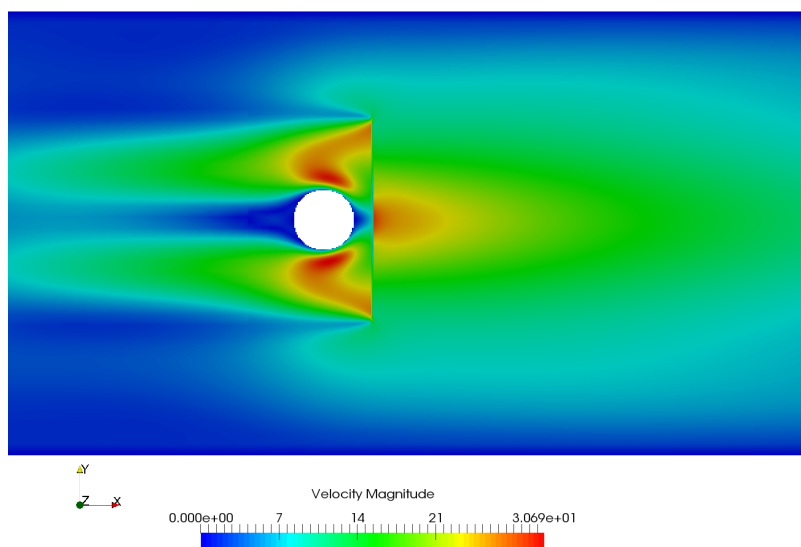


FIGURE 4.23 – Champ de vitesse adjointe

La figure 4.22 nous montre le champ de vitesse adjoint. Nous reviendrons sur sa signification dans la section 5.2. Nous pouvons tout de même noter sur la figure 4.24 qu'il n'y a pas de discontinuité au niveau des interfaces de maillage non plus dans le

calcul adjoint et que la discontinuité verticale visible sur les images est liée au terme source adjoint utilisé dans ce cas test, qui sera décrite plus en détail dans le section 5.2.

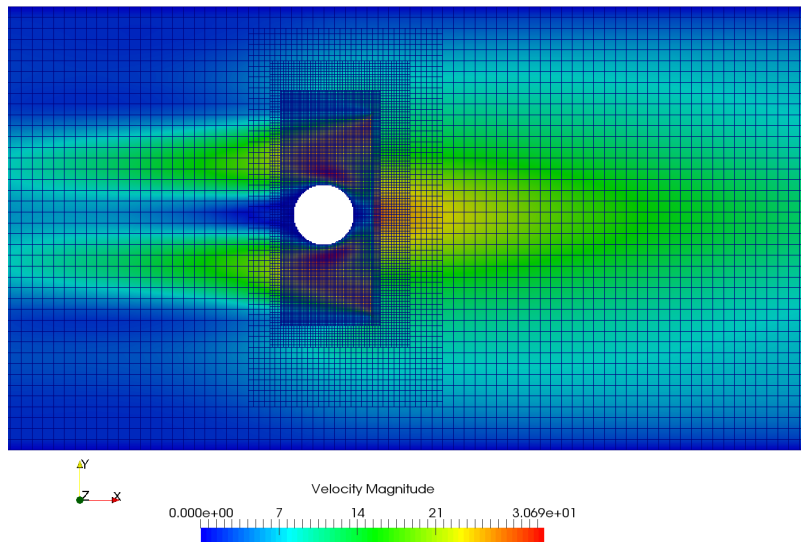


FIGURE 4.24 – Superposition du maillage et du champ de vitesse adjointe

Cette section nous a permis de montrer que les algorithmes utilisés pour les raffinements de maillage, ainsi que les opérations mathématiques effectuées pour obtenir leur formulation adjointe, sont correctes et que nous pouvons les utiliser dans un cas industriel.

Nous reviendrons sur ce cas plus en détail dans la suite du manuscrit (section 5.2).

4.6 Conclusion

Ce chapitre a permis d'étudier différents algorithmes de paroi, et de cerner le plus adapté à notre étude. L'algorithme de paroi "bounce-back" est attractif pour sa simplicité, mais il ne fait pas intervenir la distance à la paroi ce qui rend le calcul du gradient nul dans le cas où on calcule la force de traînée dans le sillage (puisque $\frac{\partial I_0}{\partial \alpha} = 0$ et $\frac{\partial R}{\partial \alpha} = 0$). L'algorithme du bounce-back a donc permis d'établir la version adjointe du bounce-back, qui est utilisée dans l'interpolation de Ginzburg.

L'algorithme de paroi de ProLB a ensuite été utilisé car nous envisageons de l'utiliser pour des cas industriels. Une interrogation est alors apparue quant à l'intérêt de dériver cette formulation pour avoir l'algorithme de paroi adjoint correspondant. En effet, cette algorithm étant très complexe, l'hypothèse suivante a été testée puis retenue : il est possible d'obtenir des gradients corrects, du bon signe et du bon ordre de grandeur, en

utilisant l'algorithme de paroi de ProLB pour le calcul direct et le link bounce-back pour le calcul adjoint. Dans ce cas là, l'algorithme de paroi adjoint n'est pas la dérivée exacte de l'algorithme de paroi direct, mais cela permet de simplifier grandement la démarche et d'obtenir des résultats concluants.

Enfin, l'interpolation de Ginzburg, ainsi que sa formulation adjointe ont donné des résultats conformes aux attentes. La formulation adjointe est relativement simple à obtenir et à implémenter dans le code adjoint. Elle fait intervenir simplement la distance à la paroi, et le calcul du gradient par l'adjoint est donc facilement réalisable.

Pour la suite des études, nous pouvons donc envisager la démarche suivante : utiliser l'algorithme de paroi de ProLB dans le calcul direct car il est le plus précis et le mieux adapté à des problèmes industriels, et utiliser l'interpolation de Ginzburg adjointe pour le problème adjoint car cela permet de calculer le gradient simplement et donne des résultats concluants dans des cas tests académiques. Le principal avantage de cet algorithme est qu'il fait intervenir simplement la distance à la paroi contrairement au link bounce back. Il reste à vérifier que cette démarche est valable dans un cas industriel, ce que nous vérifierons dans la dernière partie.

Finalement, les algorithmes utilisés pour les raffinements de maillage, dans le problème direct et dans celui adjoint, ont été validés dans la dernière section de ce chapitre. En effet, l'écoulement n'est pas discontinu au niveau des interfaces de maillage, que ce soit dans le problème direct ou adjoint.

Chapitre 5

Choix de la fonctionnelle de coût

Le choix de la fonctionnelle de coût à optimiser est une étape importante, puisque selon la formulation choisie, le calcul du gradient sera plus ou moins complexe. Une façon simple d'illustrer cela est de s'intéresser à la force de traînée. Comme nous l'avons vu dans la section bibliographie 1.2.1, deux formulations sont possibles : soit on calcule la traînée en intégrant des efforts de pression et des efforts visqueux directement sur l'obstacle en question, soit on intègre des termes de vitesses et pression dans le sillage de l'obstacle. Les deux approches ont été testées ici, la démarche adoptée pour chaque approche et les résultats qui en découlent seront présentés dans cette partie.

5.1 Force de traînée : Intégration sur l'obstacle

L'approche présentée ici correspond au calcul des efforts sur la surface de l'obstacle directement. Nous étudions cette approche dans un premier temps car elle est utilisée dans le logiciel ProLB pour le calcul de la traînée.

5.1.1 Ecoulement 2D stationnaire autour un cylindre

Problème direct

Pour étudier l'approche du calcul de la traînée proche paroi, nous étudions l'écoulement laminaire autour d'un cylindre en 2D comme le montre la figure 5.1. Le maillage est uniforme, avec une taille de maille de 2 cm, et un nombre total de mailles égal à 19907. Le nombre de Reynolds est égal à 15, le nombre de Mach à 0.003, et nous effectuons le calcul sur 4000 pas de temps avec $dt = 3.4e^{-5}s$.

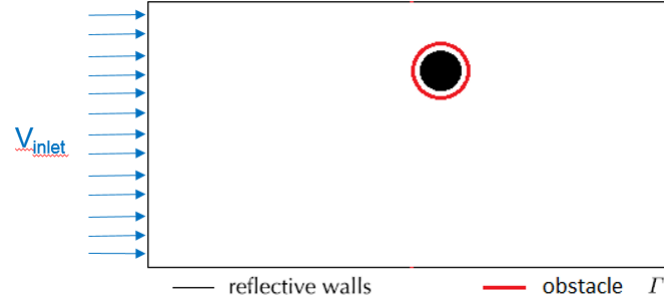


FIGURE 5.1 – Mise en données du cas 2D

Dans un premier temps, nous reprenons la fonction coût étudiée dans le cas 4.4.1 c'est-à-dire que nous considérons seulement les efforts de pression sur l'obstacle pour simplifier la démarche et mesurer sa complexité. Pour rappel :

$$I_0 = - \sum_k (p_k \vec{n}_f \cdot \vec{x} \bar{a}_f) \quad (5.1)$$

avec $k \in \Gamma$ (première couche de maillage entourant l'obstacle) , p_k la pression de cette maille, \bar{a}_f l'aire pondéré de la face de l'obstacle, \vec{n}_f le vecteur normal à la face de l'obstacle et orienté vers l'extérieur de l'obstacle, et \vec{x} l'axe horizontal puisque l'on souhaite minimiser les efforts de pression selon cet axe. Nous avons supprimé la somme en temps de 4.29 car nous nous intéressons à un écoulement stationnaire ici.

En résumé, la fonction coût est similaire à celle du cas 4.4.1, mais le paramètre d'optimisation α est désormais n'importe quelle coordonnée de n'importe quel nœud du maillage surfacique de l'obstacle.

L'écoulement étudié est laminaire et le modèle de collision BGK est utilisé, ainsi que l'interpolation de Ginzburg à la paroi du cylindre.

Gradient incomplet

La méthode du gradient incomplet a été expliquée dans la sous-section 1.2.2. Pour calculer le gradient incomplet dans notre cas, il faut calculer la dépendance explicite de la fonction coût I_0 par rapport au paramètre d'optimisation soit $\frac{\partial I_0}{\partial \alpha}$. Pour cela, nous pouvons nous servir du schéma suivant :

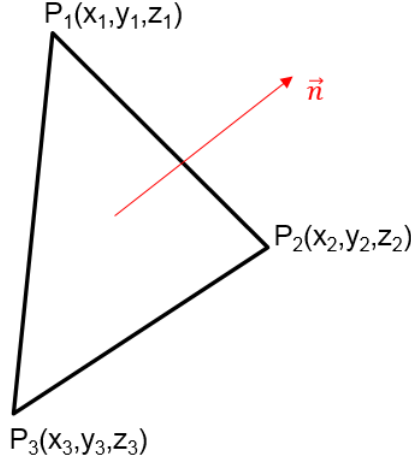


FIGURE 5.2 – Triangle utilisé dans le maillage de l'obstacle

Dans ce schéma, le triangle représente une des mailles de l'obstacle, et le paramètre d'optimisation $\alpha \in [x_1, y_1, \dots, z_3]$. Nous pouvons alors écrire :

$$\vec{n} = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \quad (5.2)$$

$$a_f = \frac{1}{2} \|\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}\| \quad (5.3)$$

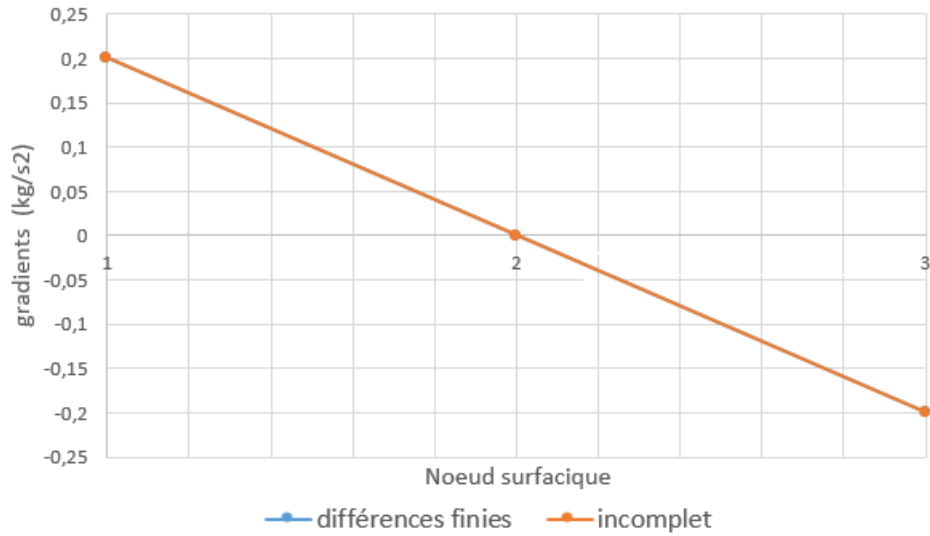
Sachant que la fonctionnelle de coût I_0 s'écrit : $I_0 = -\sum_k (p_k \vec{n}_f \cdot \vec{x} \overline{a_f})$, et que l'on suppose que l'écoulement reste inchangé quand on déplace un nœud (donc la pression ne dépend pas de α), on peut écrire :

$$\frac{\partial I_0}{\partial \alpha} = -\sum_k p_k \left(\frac{\partial (\vec{n}_f \cdot \vec{x} \overline{a_f})}{\partial \alpha} \right) \quad (5.4)$$

L'équation précédente se résume à un calcul géométrique (pour trouver $\vec{n}_f \cdot \vec{x} \overline{a_f}$) suivi de dérivées qui s'écrivent simplement par rapport à $\alpha \in [x_1, y_1, \dots, z_3]$ pour trouver le gradient incomplet. Nous allons dans un premier temps calculer les gradients incomplets dans ce cas pour les comparer aux gradients par différences finies. Cela permettra de voir si cette méthode approchée de calcul des gradients est suffisante pour effectuer une optimisation surfacique précise.

5.1.2 Résultats

Dans un premier temps, on compare les gradients incomplets aux gradients par différence finie, obtenus grâce à un schéma centré en déplaçant un nœud puis en refaisant un calcul pour chaque nœud déplacé. Nous réalisons la comparaison sur seulement 3 points pour l'instant, puis nous poursuivrons la démarche si elle semble prometteuse.



On remarque ici que les gradients incomplets sont parfaitement superposés aux gradients par différences finies. Ce premier calcul nous permet donc de valider le calcul du gradient incomplet. Cependant, Mohammadi ([57]) a montré dans son article que pour le cas qu'il étudie, en certains points de l'obstacle (bord d'attaque ou bord de fuite), le gradient incomplet est faux et donc inutilisable. En fonction du cas étudié, le gradient incomplet n'est donc pas suffisant pour fournir les directions de descente de la fonction coût. Cela constitue un point bloquant et la méthode simplifiée du calcul du gradient par le gradient incomplet n'est donc pas envisageable pour une application industrielle précise, puisque trop dépendante du cas étudié.

Une autre difficulté vient du fait que notre méthode de calcul des efforts de pression proche paroi (somme sur la première couche de mailles) n'est pas assez mature et diffère de celle de ProLB (somme sur l'obstacle directement) ce qui donne des résultats différents dans certains cas. De plus la méthode de calcul de la traînée sur l'obstacle directement dépend grandement du raffinement du maillage surfacique et volumique, mais aussi des interpolations utilisées à la surface de l'obstacle comme il a déjà été montré dans la littérature sur le sujet ([92]). Pour se libérer de ces contraintes, nous proposons d'utiliser la méthode d'intégration dans le sillage pour laquelle aucune interpolation n'est utilisée. De plus le calcul de la traînée ne dépend plus directement du raffinement du maillage surfacique de l'obstacle. Cette seconde méthode donne en outre un gradient incomplet nul comme nous le verrons dans la partie suivante, ce qui permettra de vérifier la validité de l'adjoint plus facilement.

5.2 Force de traînée : Intégration sur un volume de contrôle

Nous étudions à présent l'approche qui consiste à calculer la traînée dans le sillage grâce aux valeurs de pression et vitesse calculées. La géométrie étudiée diffère légèrement du cas précédent puisque nous rajoutons des raffinements de maillage dans cette section. Cela nous permettra de vérifier la validité des gradients en présence de zones de maillage de taille différente.

5.2.1 Ecoulement 2D stationnaire autour d'un cylindre

Problème direct

Dans cette approche, les efforts aérodynamiques sont calculés dans le sillage de l'obstacle grâce à la formule suivante 1.25 qui s'écrit pour rappel :

$$F_x = -\frac{\rho u_0^2}{2} \iint_{S_2} \left(1 - \frac{u_x}{u_0}\right)^2 dS_2 + \frac{\rho u_0^2}{2} \iint_{S_2} \left(\frac{u_y^2}{u_0^2} + \frac{u_z^2}{u_0^2}\right) dS_2 + \iint_{S_2} (p_{t0} - p_t) dS_2 \quad (5.5)$$

Le cas test présenté ici est un écoulement laminaire et stationnaire autour d'un cylindre de 22 cm de diamètre. Le nombre de Reynolds est égal à 33, le nombre de Mach à 0.33. Le domaine de calcul est quasi 2D (avec 4 mailles dans la direction du 3D) et il y a 4 zones de maillage (figure 5.4), la taille de maille minimale étant de 5 mm et le nombre total de mailles est égal à 449 496 . Le nombre total de pas de temps est 10 000. Le modèle de collision BGK est utilisé, ainsi que l'interpolation de Ginzburg à la paroi du cylindre, et les modèles de raffinement de maillage de la partie 4.5.

Dans le cas d'écoulement autour d'obstacles, la condition d'entrée du domaine de calcul est généralement une vitesse imposée et pression libre, et la condition de sortie est généralement une pression imposée et une vitesse libre. Mais nous avons supposé ici que fixer la vitesse et la pression en entrée et en sortie du domaine de calcul est valable si l'objet est suffisamment loin de l'entrée et de la sortie. Cela simplifie la mise en oeuvre des conditions limites du problème adjoint puisque nous avons dans ce cas $f^* = 0$ en entrée et sortie.

Concernant la condition initiale, nous avons $f^*(\mathbf{x}, t_f) = 0$ comme expliqué dans la partie 4.1.

Problème adjoint

Pour calculer le terme source à implémenter dans le solveur adjoint, qui correspond à cette définition de la force de traînée, on peut utiliser dans un premier temps les résultats suivants :

$$\frac{\partial \rho}{\partial f_i} = 1 \quad (5.6)$$

$$\frac{\partial \mathbf{j}}{\partial f_i} = \mathbf{c}_i \quad (5.7)$$

$$\frac{\partial \mathbf{u}}{\partial f_i} = \frac{1}{\rho}(\mathbf{c}_i - \mathbf{u}) \quad (5.8)$$

En combinant les équations (5.6), (5.7) et (5.8), on obtient :
pour $k \in S_2$:

$$\begin{aligned} \frac{\partial I_0}{\partial f_i} = & c_s^2 + (u_x c_{ix} + u_y c_{iy} + u_z c_{iz}) - \frac{u_x^2 + u_y^2 + u_z^2}{2} \\ & - \frac{(u_y(2c_{iy} - u_y)) + (u_z(2c_{iz} - u_z))}{2} + \frac{u_0^2}{2} - c_{ix} u_0 + \frac{u_x}{2}(2c_{ix} - u_x) \end{aligned} \quad (5.9)$$

sinon :

$$\frac{\partial I_0}{\partial f_i} = 0 \quad (5.10)$$

Nous avons donc trouvé le terme source à implémenter dans le modèle de collision adjoint BGK. D'autre part, l'interpolation adjointe de Ginzburg est utilisée à la paroi du cylindre, ainsi les modèles de raffinement de maillage de la partie 4.5.

On remarque ici qu'il n'y a pas de dépendance explicite entre la fonction coût et le paramètre d'optimisation, donc $\frac{\partial I_0}{\partial \alpha} = 0$. Cela signifie que la première partie du gradient, que l'on a appelé gradient incomplet, est nul dans ce cas. Cela simplifie donc grandement le calcul du gradient puisqu'il suffit ici de calculer le gradient par adjoint pour obtenir le gradient total (équation 5.14).

Gradient par l'adjoint

Le calcul du gradient par l'adjoint correspond à la deuxième partie de l'équation du calcul du gradient, c'est-à-dire $f^* \frac{\partial R}{\partial \alpha}$. Nous verrons dans ce paragraphe comment calculer ce terme.

Tout d'abord, nous faisons un calcul stationnaire, ce qui signifie que l'on utilise les valeurs finales des calculs direct et adjoint pour calculer le gradient. La dépendance temporelle du gradient disparaît donc on peut écrire :

$$\nabla I_0 = \sum_k \sum_i f^* \frac{\partial R}{\partial \alpha} \quad (5.11)$$

Il reste donc deux sommes dans le calcul du gradient, une sur i : les directions de propagation (on utilise le réseau D3Q19 donc il y a 19 directions), et une autre somme

sur k : les nœuds du domaine de calcul. Nous voulons donc calculer le terme restant : $\frac{\partial R}{\partial \alpha}$. R étant une équation d'état de la LBM, cela peut être soit l'équation en cœur de fluide, soit un raffinement de maillage, soit l'algorithme de paroi. Sachant que α est n'importe quelle coordonnée de n'importe quel nœud du maillage surfacique de l'obstacle, ce paramètre n'agit que sur ce qui est modélisé proche de la paroi. Proche paroi, on utilise l'interpolation de Ginzburg, donc nous devons calculer la dépendance de l'interpolation de Ginzburg à n'importe quelle coordonnée de n'importe quel nœud du maillage surfacique.

On rappelle que l'interpolation de Ginzburg s'écrit :

$$R_j = f_j(\mathbf{x}_k, t_{n+1}) - f_{opp(j)}^{col}(\mathbf{x}_k, t_n) - \frac{1 - 2d}{1 + 2d} [f_{opp(j)}^{col}(\mathbf{x}_{k+1}, t_n) - f_j^{col}(\mathbf{x}_k, t_n)] \quad (5.12)$$

Dans cette équation, le seul terme qui dépend de α est la distance d à la paroi parce que les fonctions de distribution n'en dépendent pas. Pour calculer $\frac{\partial R}{\partial \alpha}$, il faut donc d'abord calculer $\frac{\partial d}{\partial \alpha}$. Pour cela, la figure 5.3 permet de mieux visualiser et comprendre le calcul effectué.

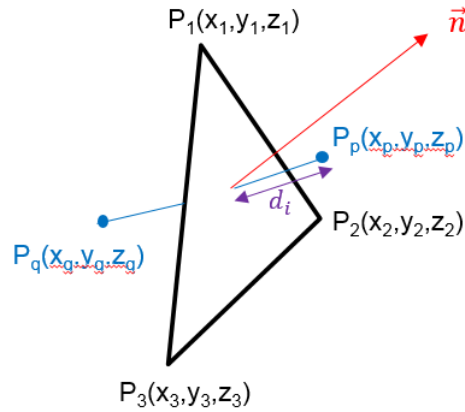


FIGURE 5.3 – Représentation de la distance d entre un nœud fluide et une paroi solide

Si l'on suppose que $P_1P_2P_3$ est un triangle du maillage surfacique (l'obstacle est maillé avec un maillage triangulaire, et la paramètre d'optimisation est $\alpha \in \{x_1, x_2, \dots, z_3\}$), P_P est un nœud fluide, et P_Q est un nœud fluide fictif qui serait le nœud correspondant au bout du lien i , et \vec{n} est la normale au triangle, nous pouvons calculer la distance d en utilisant la formule géométrique suivant :

$$d_i = \|\overrightarrow{P_p P_q}\| \frac{\overrightarrow{P_1 P_p} \cdot \vec{n}}{\overrightarrow{P_1 P_p} \cdot \vec{n} - \overrightarrow{P_1 P_q} \cdot \vec{n}} \quad (5.13)$$

Une fois que l'on a calculé d , sa dérivation par rapport à α est faite assez facilement.

Nous obtenons donc l'équation finale du gradient par l'adjoint :

$$\nabla I_0 = \sum_k \sum_i f_i^*(\mathbf{x}_k, t) \frac{4 \frac{\partial d}{\partial \alpha}}{(1 + 2d)^2} [f_{opp(i)}^{col}(\mathbf{x}_{k+1}, t) - f_i^{col}(\mathbf{x}_k, t)] \quad (5.14)$$

Nous pouvons donc remarquer qu'avec la connaissance du champ direct $f_{opp(i)}^{col}(\mathbf{x}_{k+1}, t)$ et $f_i^{col}(\mathbf{x}_k, t)$, ainsi que du champ adjoint $f_i^*(\mathbf{x}_k, t)$, nous pouvons calculer les gradients surfaciques indépendamment du nombre de paramètres d'optimisation, ce qui constitue le principal intérêt de la méthode adjointe.

5.2.2 Résultats

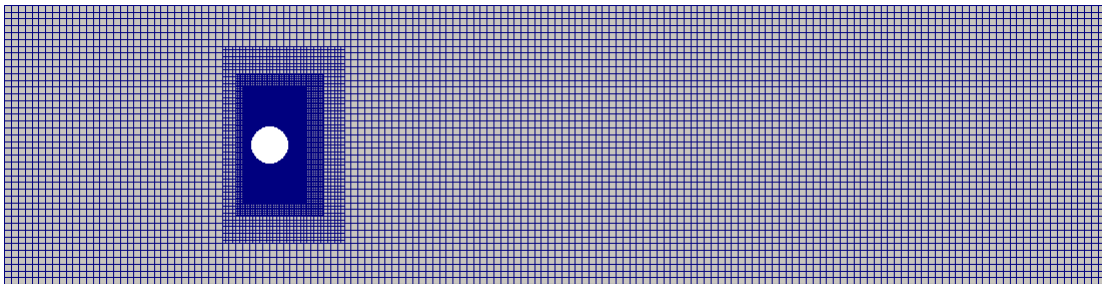


FIGURE 5.4 – Maillage du cas 2D stationnaire

Le calcul est effectué jusqu'à convergence et obtention d'un état stationnaire, dont le champ de vitesses est montré sur la figure suivante :

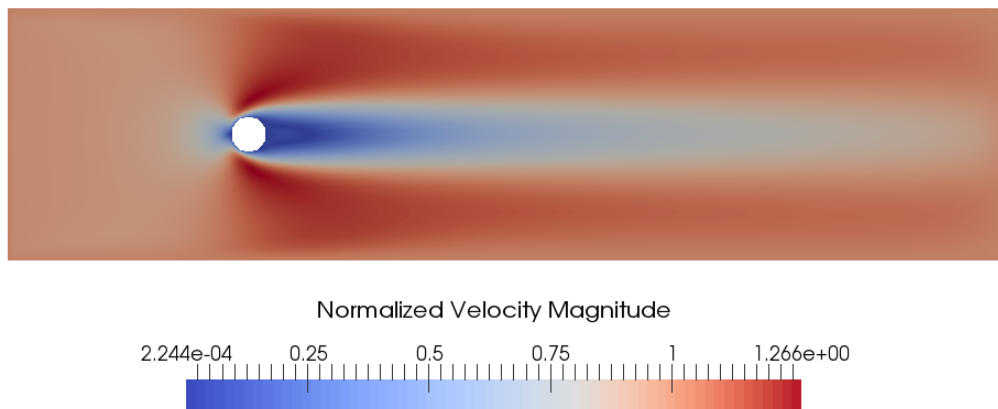


FIGURE 5.5 – Champ de vitesse autour du cylindre

Le calcul adjoint est ensuite effectué jusqu'à ce qu'on obtienne un état convergé et stationnaire aussi :

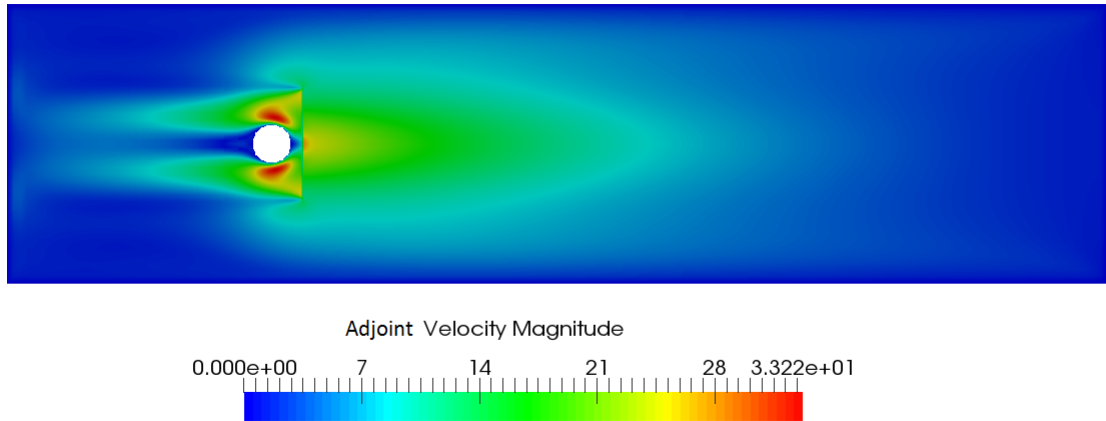


FIGURE 5.6 – Champ de vitesse adjointe autour du cylindre

On peut voir sur la figure 5.6 que le terme source du modèle adjoint apparaît clairement. Cela correspond à la ligne où la force de traînée a été calculée dans le calcul direct. Nous pouvons aussi remarquer qu'il n'y a pas de discontinuité à travers les interfaces de maillage de taille différente.

Les valeurs convergées des problèmes direct et adjoint sont ensuite utilisées pour calculer les sensibilités à la surface du cylindre grâce à l'équation 5.14. Les sensibilités, ou gradients, peuvent être visualisés sous forme de vecteurs (nous représentons $-\nabla I_0$ ce qui permet de mieux visualiser la déformation : on optimise en déformant suivant les flèches) :

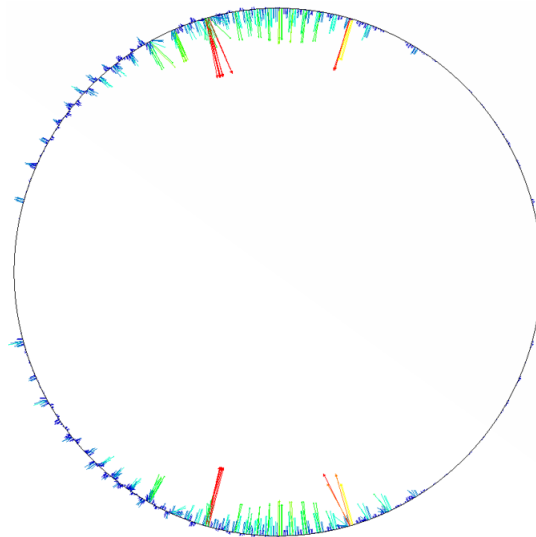


FIGURE 5.7 – Sensibilités sur le cylindre

Les sensibilités sont ensuite comparées à celles obtenues avec la méthode des différences finies pour valider le modèle adjoint, sur plusieurs nœuds du maillage du cylindre présentés sur la figure 5.8.

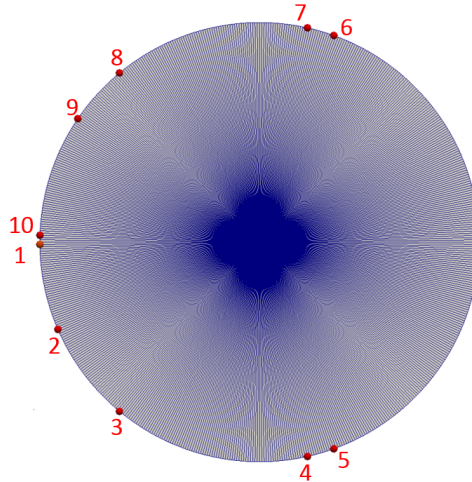


FIGURE 5.8 – Nœuds surfaciques utilisés pour la comparaison

Un schéma centré est utilisé pour les différences finies. Une étude de convergence est menée sur le pas du schéma des différences finies, avec les pas suivants : 10^{-3} m, 10^{-4} m, 10^{-5} m, and 10^{-6} m.

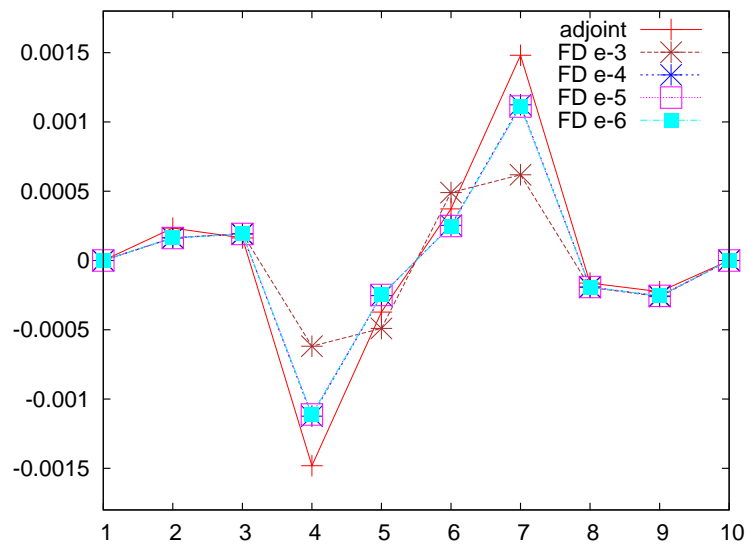


FIGURE 5.9 – Comparaison des gradients par adjoint et par différences finies

Les sensibilités calculées par l'adjoint sont proches de celles par différences finies,

même si l'écart sur certains points est plus élevé que sur d'autres. Les gradients les plus forts, pour lesquels les écarts sont les plus remarquables, sont ceux des nœuds situés proches des extrémités hautes et basses du cylindre. Cependant, nous pouvons remarquer dans le travail de Nadarajah [60] et de Li [49] qu'ils trouvent aussi des écarts entre leurs gradients par adjoint et par différences finies, surtout pour les valeurs élevées de gradient. Une explication de ces erreurs est proposée par Anderson et Venkatakrishnan [6]. Le point de départ est de considérer que la fonction coût ne dépend pas seulement des variables d'état (f) et des paramètres d'optimisation (α), mais aussi du maillage volumique fluide (X) lorsqu'on considère un problème discrétisé (ce qui est le cas ici) : $I_0 = I_0(f, \alpha, X)$. Cela signifie que lorsqu'on calcule le gradient en utilisant la méthode adjointe, un terme supplémentaire apparaît dans le calcul du gradient comparé à l'équation 2.47 :

$$\nabla I_0 = \frac{\partial I_0}{\partial \alpha} + f^* \left(\frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial X} \frac{\partial X}{\partial \alpha} \right) \quad (5.15)$$

Le terme $\frac{\partial X}{\partial \alpha}$, nommé "grid sensitivities" dans leur travail, représente la sensibilité des nœuds du maillage volumiques fluides par rapport à des variations des paramètres d'optimisation. Ici, les paramètres d'optimisation sont les coordonnées des nœuds du maillage surfacique du cylindre, donc $\frac{\partial X}{\partial \alpha}$ représente comment le fluide volumique fluide est modifié lorsque la position des nœuds du maillage du cylindre varie. Ce terme est difficile à calculer dans notre cas, mais il peut être négligé si la taille du maillage surfacique est suffisamment petite. En effet, si le maillage du cylindre est suffisamment raffiné, les variations de position des nœuds du maillage surfacique n'auront pas beaucoup d'effets sur le maillage fluide environnant. Dans le cas présenté ici, le cylindre est maillé avec 600 nœuds, donc nous pouvons raisonnablement supposer que les erreurs observées sur la figure 5.9 sont dues au terme $\frac{\partial X}{\partial \alpha}$ qui a été négligé. De plus, Anderson observe que les "grid sensitivities" ne disparaissent pas lorsque le maillage est raffiné sur les nœuds présentant des singularités, tels que les nœuds sur lesquels d'importants gradients de vitesse ont lieu par exemple. Dans son cas, la singularité qu'il évoque est localisée sur le bord de fuite du profil d'aile qu'il étudie. Dans le cas présenté ici, les singularités sont localisées proches des points en haut et en bas du cylindre, où de larges gradients de vitesse apparaissent dues aux zones de décollement. Cela peut expliquer les erreurs observées sur les nœuds 4 et 7 sur la figure 5.9.

Dans une utilisation industrielle du solveur adjoint, la surface du véhicule est maillée de façon raffinée, donc nous pouvons émettre l'hypothèse que le terme $\frac{\partial X}{\partial \alpha}$ est négligeable sur la plus grande partie du véhicule. De plus, nos erreurs étant du même ordre de grandeur que celles de Nadarajah [60] et de Li [49], nous avons décidé de valider le modèle adjoint pour le reste de l'étude. Enfin, l'algorithme d'optimisation tend à réduire la surface frontale du cylindre, pour qu'il devienne plus profilé, ce qui est logiquement attendu pour réduire la force de traînée.

La dernière étape dans le cycle d'optimisation est de déformer le cylindre en utilisant une méthode de descente à pas constant. Nous effectuons 3 cycles d'optimisation ici, et

l'évolution de la forme du cylindre est visible sur la figure 5.10. Nous pouvons remarquer que le cylindre devient de plus en plus profilé après chaque cycle d'optimisation.

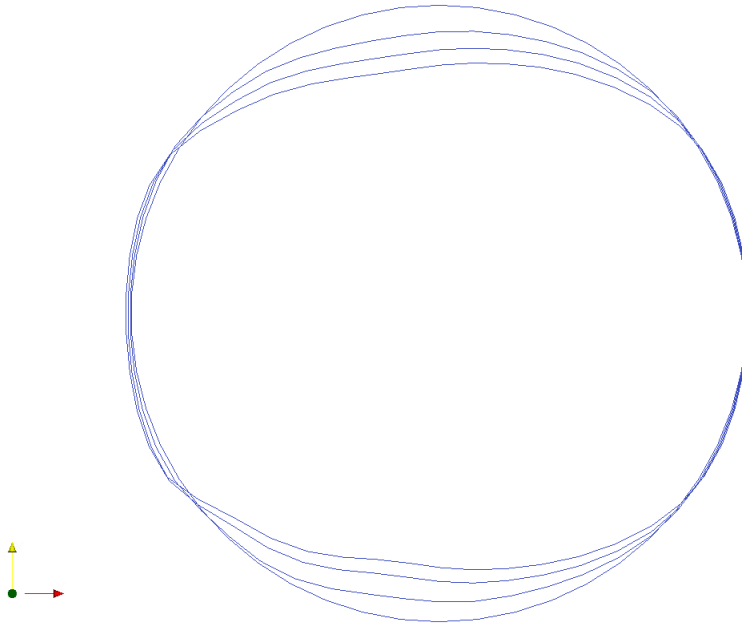


FIGURE 5.10 – Déformation du cylindre après 3 cycles d'optimisation

L'écoulement est ensuite recalculé pour chaque forme :

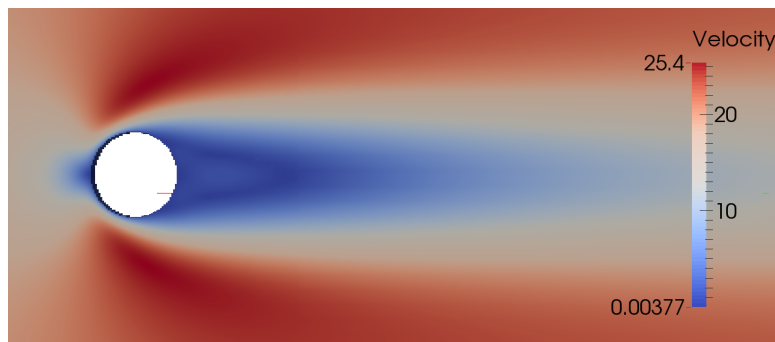


FIGURE 5.11 – Champ de vitesse derrière le cylindre initial

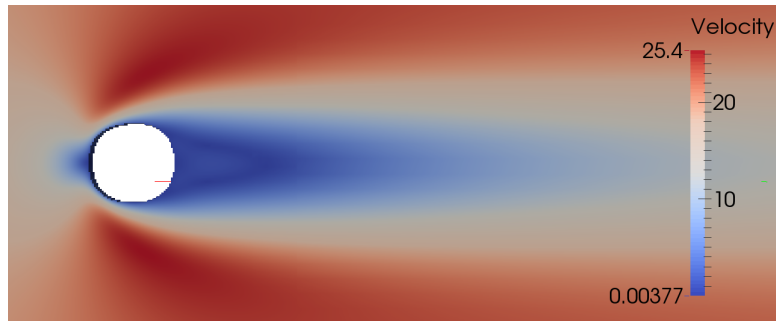


FIGURE 5.12 – Cycle 1 : Champ de vitesse derrière le cylindre optimisé

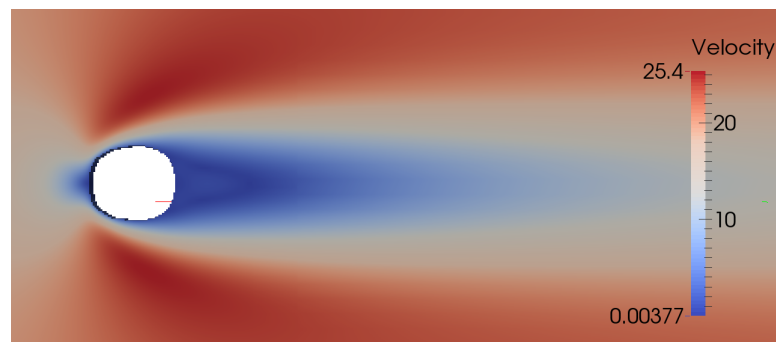


FIGURE 5.13 – Cycle 2 : Champ de vitesse derrière le cylindre optimisé

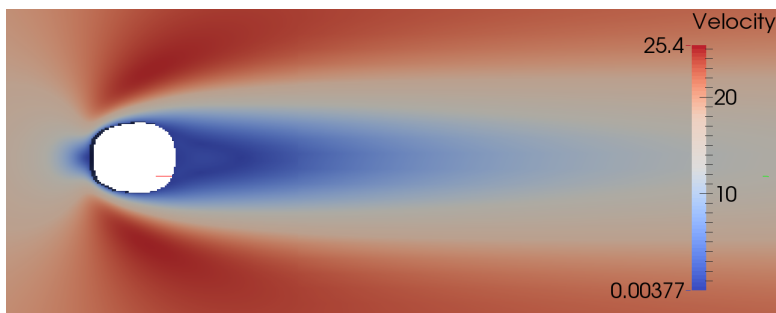


FIGURE 5.14 – Cycle 3 : Champ de vitesse derrière le cylindre optimisé

Nous pouvons voir que la taille du sillage est réduite après chaque cycle d'optimisation. Finalement, la force de traînée est réduite de plus de 12% après 3 cycles d'optimisation :

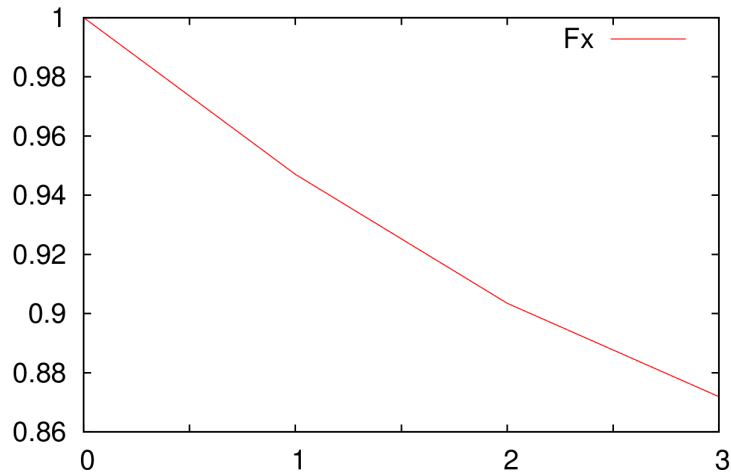


FIGURE 5.15 – Réduction de la force de traînée (adimensionnée) sur le cylindre

5.3 Conclusion

En conclusion de ce chapitre, le calcul de la force de traînée dans le sillage a été retenu. En effet, cela simplifie le calcul du gradient car le gradient incomplet devient nul, et cela permet aussi de s'affranchir des problèmes d'interpolation au niveau de l'obstacle ainsi que de taille de maillage. Cette méthode semble donc plus robuste, et mieux adaptée à notre problème. Elle a donné des résultats concluants pour un cas test 2D, et son application à un cas industriel est directe. Même si la formule d'Onorato suppose un certain nombre d'hypothèses (écoulement incompressible, efforts moyennés en temps), sa simplicité d'implémentation justifie son choix. De plus, cette formule permet de décomposer la traînée en trois composantes (perte de pression, tourbillons longitudinaux et tourbillons transversaux), chacune présentant un intérêt pour le secteur automobile comme nous le verrons dans le cas industriel.

Chapitre 6

Choix de l'approximation temporelle

Le chapitre concluant cette seconde partie concerne la dépendance temporelle du problème direct, à savoir s'il est stationnaire ou bien instationnaire. En effet, si le calcul direct est stationnaire, l'enregistrement de la dernière itération permet de calculer un adjoint lui aussi stationnaire. Par contre, si le calcul direct est instationnaire, plusieurs possibilités existent pour le calcul adjoint, qui diffèrent quant à la précision des résultats mais aussi à l'effort de calcul (section 3.5).

La question principale à laquelle répond ce chapitre est donc : sous quelle hypothèse peut-on obtenir un calcul adjoint et des gradients précis pour un cas instationnaire, tout en respectant un coût de calcul raisonnable pour un calcul industriel ?

Pour répondre à cette question, nous étudierons deux méthodes évoquées dans la partie bibliographique 3.5 : la méthode "full unsteady" et la méthode "time-averaged flow - steady adjoint". La comparaison des résultats, sur un cas test 2D, nous permettra de nous positionner sur la démarche à adopter pour le reste des études.

6.1 Direct instationnaire - Adjoint instationnaire

La première méthode que l'on présente ici ("full unsteady") repose sur l'enregistrement complet du champ direct instationnaire pour avoir un adjoint instationnaire.

6.1.1 Ecoulement 2D instationnaire autour d'un cylindre

Problème direct

Pour étudier le problème de l'instationnarité de l'écoulement direct, nous choisissons de simuler l'écoulement derrière un cylindre pour un Reynolds égal à 125. Pour ce nombre de Reynolds, il a été montré qu'un échappement tourbillonnaire périodique se produit

derrière le cylindre ([78]). Ce cas test est donc idéal pour étudier l'instationnarité de l'écoulement puisque nous pouvons moyenniser le calcul direct sur un cycle de lâcher de tourbillons. Le modèle de collision BGK est utilisé, ainsi que l'interpolation de Ginzburg à la paroi du cylindre et la force de traînée est calculée dans le sillage (équation 1.25). Le nombre de Mach est égal à 0.1.

Problème adjoint

Sachant que l'on réalise un calcul instationnaire, la fonction coût est définie sur un cycle de lâcher de tourbillons de la façon suivante avec N_T le nombre d'itérations sur un cycle :

$$I_0 = \frac{1}{N_T} \sum_t F_x(t) \quad (6.1)$$

Pour limiter la taille des fichiers à relire, le calcul direct n'est enregistré que sur un seul cycle de lâcher de tourbillons, entre l'itération 25806 et 26774, donc $N_T = 968$ ici. L'écoulement étant strictement périodique, toute l'information physique est contenue dans une seule période et cela suffit donc pour étudier la méthode "full unsteady".

Le gradient contient maintenant une triple somme, en espace, temps, et vitesse. Il faut ici faire attention à la somme en temps : sachant que le calcul adjoint est fait en temps inverse, on sommerait en utilisant la valeur initiale du calcul direct et la valeur finale de l'adjoint par exemple, puis la seconde valeur du direct et l'avant-dernière de l'adjoint pour la somme en temps dans le calcul du gradient.

$$\nabla I_0 = \frac{1}{N_t} \sum_n \sum_k \sum_i f_i^*(\mathbf{x}_k, t) \frac{4 \frac{\partial d}{\partial \alpha}}{(1 + 2d)^2} [J_{opp(i)}^{col}(\mathbf{x}_{k+1}, t) - f_i^{col}(\mathbf{x}_k, t)] \quad (6.2)$$

Le modèle de collision adjoint BGK est utilisé, ainsi que l'interpolation adjointe de Ginzburg à la paroi du cylindre.

Concernant le terme source dans le calcul adjoint, il est le même que dans le cas du calcul de la traînée dans le sillage (5.9), à la différence près que les valeurs de vitesse et de pression du calcul direct ne sont pas égales aux valeurs de la dernière itération. En effet dans le cas instationnaire, chaque valeur est relue à chaque itération en sens inverse, c'est-à-dire qu'à la première itération du calcul adjoint, on relit les valeurs finales de pression et vitesse (la 26774ième itération), à la seconde itération de l'adjoint, on relit l'avant dernière itération du calcul direct (l'itération 26773), et ce jusqu'à avoir parcouru toutes les itérations. Cependant une différence existe avec le cas instationnaire présenté dans [90], puisqu'il faut que le champ adjoint s'établisse avant de calculer l'état adjoint. Cela signifie que les fichiers de pression et vitesse du calcul direct sont relus entièrement plusieurs fois à la suite jusqu'à ce que l'adjoint atteigne un état périodique.

Une fois l'état périodique atteint pour le calcul adjoint (dans ce cas après 3 cycles de relecture du direct), il est possible de calculer le gradient en utilisant la formule 6.2, sur le dernier cycle de l'adjoint.

6.1.2 Résultats

Le domaine d'étude est représenté sur la figure 6.1. Le maillage est uniforme, sans raffinement, car il est préférable d'avoir un maillage de taille assez faible pour que la relecture du champ direct sur un cycle ne soit pas trop longue pour faire tourner l'adjoint. Le maillage est constitué de 19903 mailles, de taille 2 cm, avec 200 mailles en longueur et 100 mailles en largeur. Le diamètre du cylindre est de 22,38 cm.

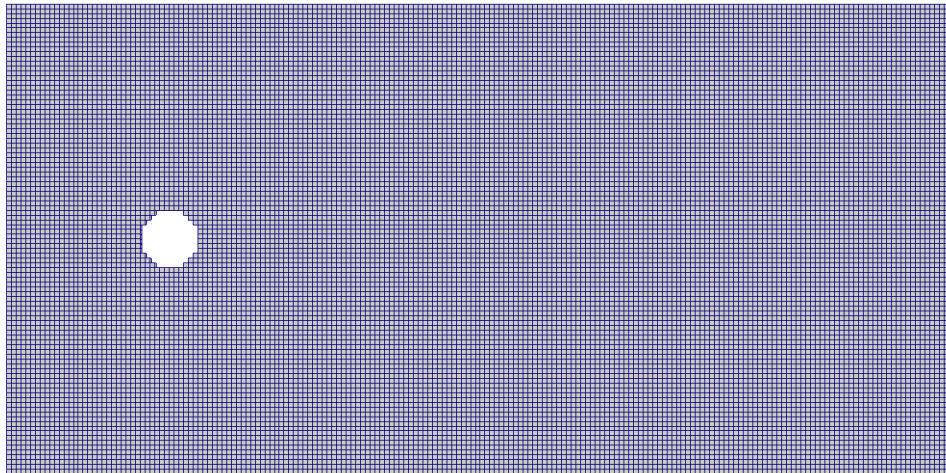


FIGURE 6.1 – Maillage du cas 2D instationnaire

Le calcul est instationnaire, et nous pouvons visualiser sur la figure 6.2 le champ de vitesse derrière le cylindre pour une itération du calcul direct.

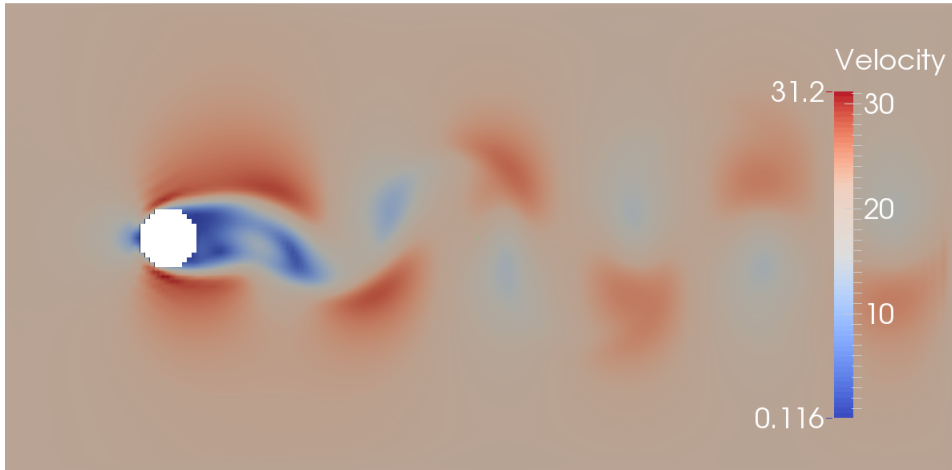


FIGURE 6.2 – Champ de vitesse instantané derrière le cylindre

Nous observons bien sur la figure 6.2 une itération en instantané du lâcher de tourbillons derrière le cylindre (avec T une période de lâcher de tourbillons).

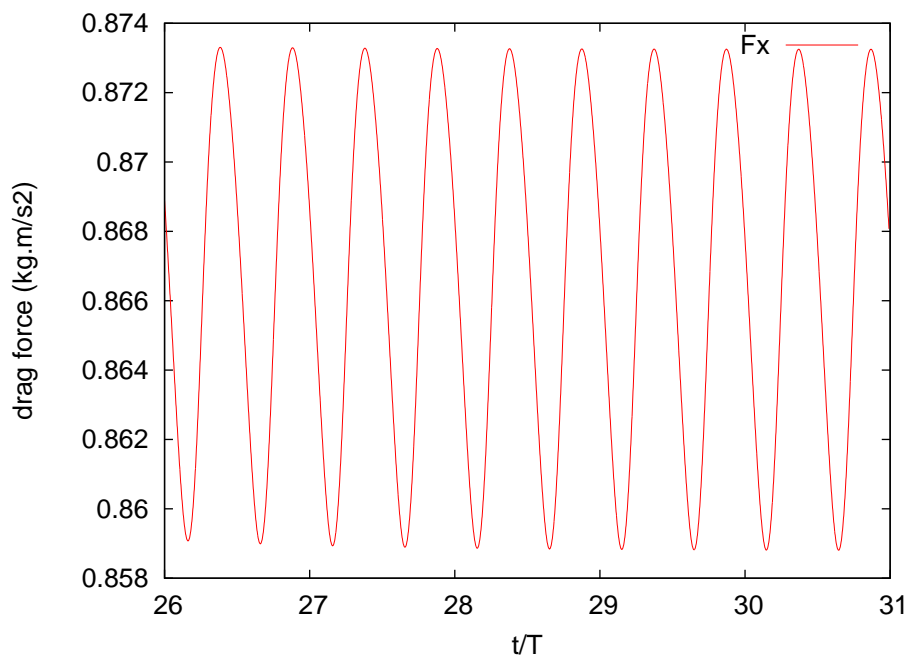


FIGURE 6.3 – Force de traînée du cas 2D instationnaire

La force de traînée calculée dans le sillage (figure 6.3) est quant à elle bien périodique comme attendu, de la même période que le lâcher de tourbillons.

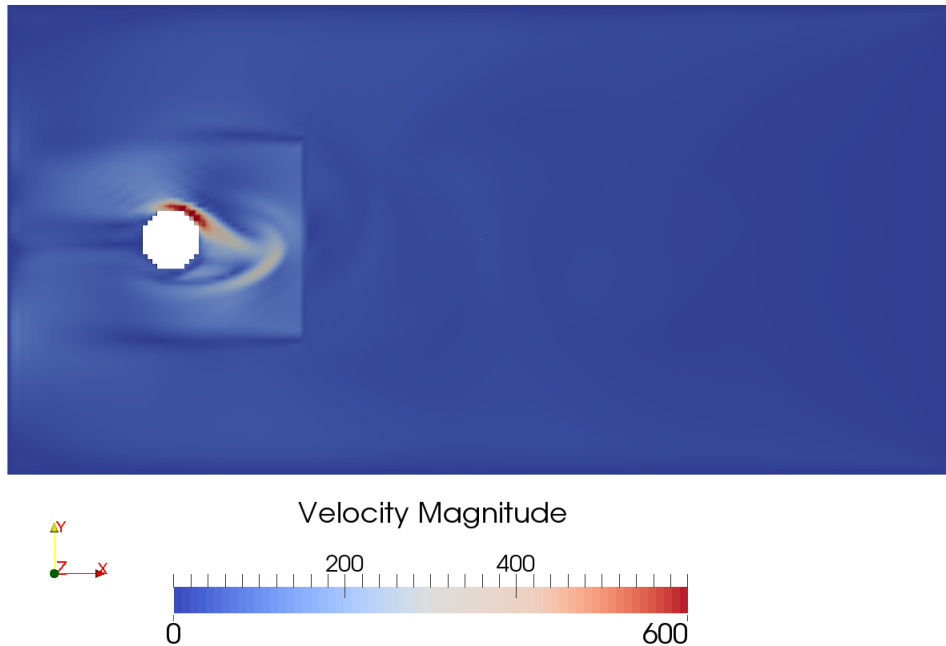


FIGURE 6.4 – Champ de vitesse adjointe instantané derrière le cylindre

Enfin, nous pouvons observer sur la figure 6.4 une itération en instantané du champ de vitesse adjointe, ainsi que le terme source qui vient bien de la ligne où a été calculée la traînée dans le calcul direct.

Les résultats du calcul du gradient sur ce cas sont présentés sur la figure 6.7, en comparaison avec la méthode "directe instationnaire moyennée - adjoint stationnaire".

6.2 Direct instationnaire moyenné - Adjoint stationnaire

La seconde méthode présentée ici consiste à moyenner le champ direct instationnaire pour obtenir un adjoint stationnaire.

6.2.1 Écoulement 2D instationnaire moyenné autour d'un cylindre

Problème direct

Dans ce cas, nous reprenons la mise en données du cas précédent, et nous moyennons le champ direct sur un cycle de lâcher de tourbillons (entre les itérations 25806 et 26774). Le champ de vitesse moyenné est représenté sur la figure 6.5.

Problème adjoint

La fonction coût est définie de la même façon que dans le cas précédent, c'est-à-dire sur un cycle de lâcher de tourbillons de la façon suivante avec N_T le nombre d'itérations sur un cycle :

$$I_0 = \frac{1}{N_T} \sum_t F_x(t) \quad (6.3)$$

Le calcul direct est ensuite moyenné sur un seul cycle, entre l'itération 25806 et 26774, donc $N_T = 968$ ici. Le gradient contient maintenant une double somme, en espace et en vitesse puisque la somme en temps a disparu. En effet, seule la valeur moyenne du calcul direct est utilisée pour calculer l'adjoint et le gradient.

$$\nabla I_0 = \sum_k \sum_i f_i^*(\mathbf{x}_k, t) \frac{4 \frac{\partial d}{\partial \alpha}}{(1 + 2d)^2} [\overline{f_{opp(i)}^{col}}(\mathbf{x}_{k+1}, t) - \overline{f_i^{col}}(\mathbf{x}_k, t)] \quad (6.4)$$

avec $\overline{f_{opp(i)}^{col}} = \frac{1}{N_T} \sum_t f_{opp(i)}^{col}(\mathbf{x}_k, t)$.

Concernant le terme source dans le calcul adjoint, il est égal à celui de l'équation : (5.9). Les valeurs de vitesse et de pression du calcul direct sont égales ici aux valeurs moyennées du champ direct. Il faut ici aussi attendre que le champ adjoint converge avant de calculer le gradient, grâce aux valeurs convergées des calculs direct et adjoint.

6.2.2 Résultats

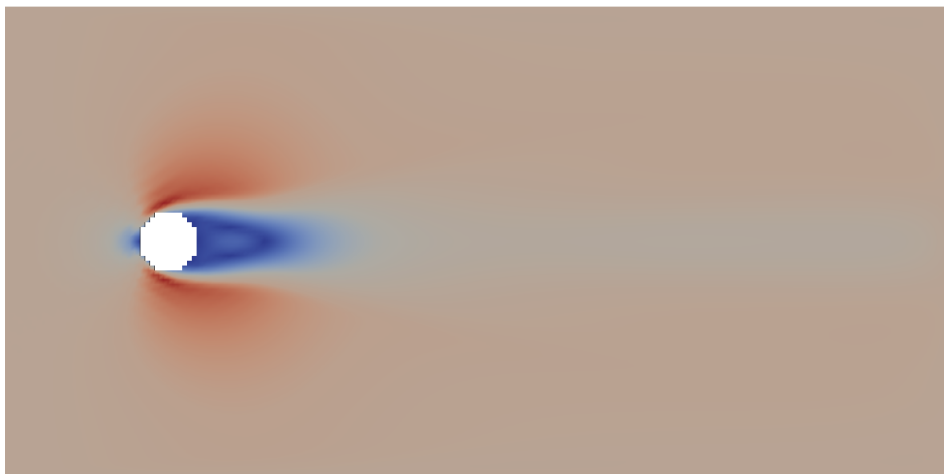


FIGURE 6.5 – Champ de vitesse moyenné du cas 2D instationnaire

Nous pouvons voir sur la figure 6.5 que le champ de vitesse moyen ne présente plus de lâcher de tourbillons ce qui est normal puisqu'on a moyenné sur un cycle.



FIGURE 6.6 – Champ de vitesse moyenné adjoint du cas 2D instationnaire

Nous pouvons voir sur la figure 6.6 que le champ de vitesse adjoint est ici stationnaire et ne présente plus de lâcher de tourbillons à partir du terme source. Nous pouvons à présent comparer les gradients obtenus avec les deux méthodes : "full unsteady" et "time-averaged flow - steady adjoint".

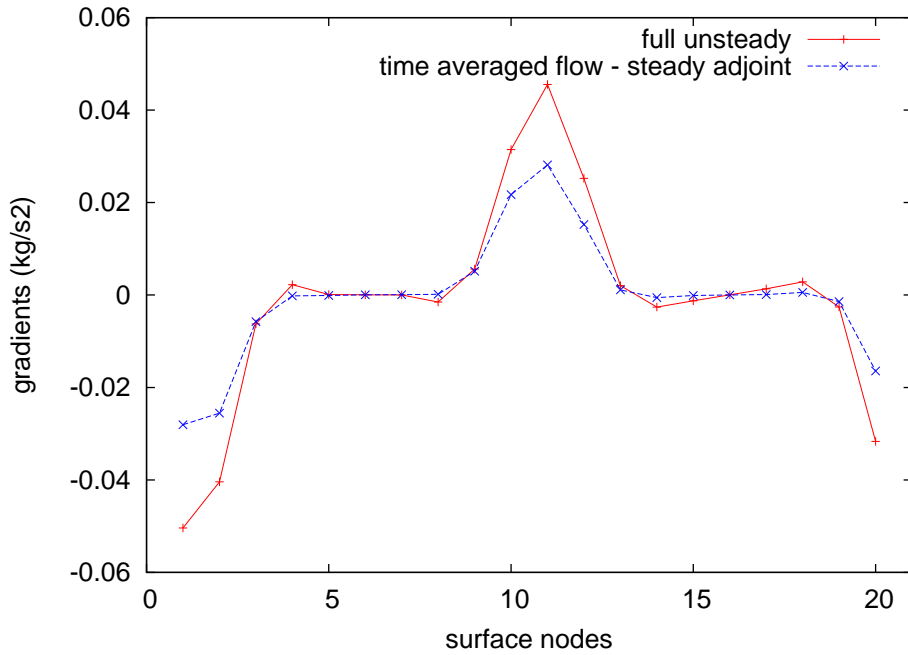


FIGURE 6.7 – Comparaison des gradients pour le cas 2D instationnaire

La figure 6.7 nous montre qu'il existe des écarts sur certains points mais nous consi-

dérons qu'ils sont acceptables pour une utilisation industrielle car les gradients obtenus avec les deux méthodes ont le même signe et le même ordre de grandeur. La méthode "time-averaged flow - steady adjoint" nous donne donc des gradients relativement corrects, à un coût de calcul très intéressant par rapport à la méthode "full unsteady".

6.3 Conclusion

Comme le montre la figure 6.7, les gradients ont des signes et des ordres de grandeurs similaires pour les méthode "full-unsteady" et "time-averaged flow - steady adjoint". Ces résultats concordent avec ceux de Nadarajah et Jameson [60], et les différences observées entre les deux méthodes sont du même ordre de grandeur dans leur étude. Cela permet donc de valider l'hypothèse selon laquelle des gradients corrects peuvent être obtenir avec la méthode "time-averaged flow - steady adjoint", qui est très intéressante en temps de calcul. De plus, l'utilisation de cette hypothèse au cas industriel est primordiale car il n'est pas envisageable d'enregistrer le calcul direct sur plusieurs centaines de milliers d'itérations sachant que le maillage volumique est très lourd dans le cas industriel.

Conclusion de la partie II

Cette deuxième partie nous a permis de faire les choix nécessaires concernant les hypothèses à retenir pour préparer une utilisation industrielle.

Dans le chapitre 4, nous avons vu que nous pouvons utiliser un algorithme de paroi différent pour le calcul direct et adjoint, et obtenir des gradients relativement précis. Cela permet d'utiliser le solveur ProLB standard avec son algorithme de paroi complexe pour le calcul direct, et d'utiliser l'interpolation de Ginzburg qui est un algorithme de paroi plus simple pour le calcul adjoint, afin de pouvoir calculer les gradients à la paroi facilement. Le chapitre 4 nous a aussi permis de montrer la validité des algorithmes de transitions de maillage pour le calcul adjoint.

Le chapitre 5 nous a montré que le calcul de la traînée dans le sillage est la méthode la plus astucieuse pour développer un calcul adjoint. En effet, cela simplifie le calcul du gradient car le gradient incomplet devient nul, et cela permet aussi de s'affranchir des problèmes d'interpolation au niveau de l'obstacle ainsi que de taille de maillage. Sachant que le calcul des efforts est actuellement fait sur la paroi du véhicule dans le solveur ProLB utilisé en projet chez Renault, la méthode proposée ici diffère de celle du solveur standard, mais cela ne constitue pas un point bloquant pour la suite. Au contraire, l'implémentation du calcul des forces aérodynamiques est faite assez simplement dans le solveur, donc cette routine peut se coder simplement.

Enfin, le chapitre 6 a permis d'étudier les approximations temporelles du problème direct utiles pour la réduction du coût de calcul de l'adjoint. Nous avons vu que moyenniser le champ direct instationnaire sur un nombre de pas de temps suffisant (une fois que le calcul a convergé), et se servir de ce champ moyen pour calculer un adjoint stationnaire, permet de réduire l'effort de calcul de façon très intéressante, tout en conservant des gradients précis. Cette hypothèse est valable dans nos cas d'étude puisque nous nous intéressons à la traînée moyennée en temps, et elle est d'autant plus importante que l'enregistrement du champ direct pour un cas industriel à grandes échelles n'est pas imaginable. Nous utiliserons donc cette hypothèse par la suite.

En résumé cette seconde partie nous a permis de développer et valider les différentes hypothèses nécessaires à un calcul industriel. Elles ont été justifiées par des exemples

concrets en prenant en compte les complexités mais aussi les simplifications qu'elles apportent.

Troisième partie

Extension des algorithmes adjoints pour les applications 3D industrielles

CONFIDENTIEL INDUSTRIE

Chapitre 7

Résultats

Le cas industriel présenté ici concerne un véhicule réel utilisé chez Renault en soufflerie, dans un écoulement très turbulent avec un nombre de Reynolds d'environ 10^7 et un nombre de Mach de 0.1. Le domaine de calcul est composé de 5 zones de maillage (figure 7.1). Le nombre total de mailles fluides est de 3 620 362, et le calcul a été fait en parallèle sur 28 cœurs sur 30 000 itérations. Le nombre de paramètres d'optimisation, c'est-à-dire le nombre de nœuds du maillage surfacique du véhicule, est de 97 271. L'intérêt de la méthode adjointe est donc ici évident sachant que le nombre de paramètres d'optimisation est très élevé. La vitesse en entrée du domaine est de 130 km/h et le fluide étudié est l'air avec une viscosité cinématique de $1.5e^{-5}m^2/s$. Nous rappelons ici que ce cas permet de valider l'utilité de la méthode adjointe. Pour cela, quelques cycles d'optimisation seront suffisants, puisque nous n'avons pas pour objectif d'optimiser complètement une forme jusqu'à convergence mais seulement à démontrer la validité de l'approche présentée ici.

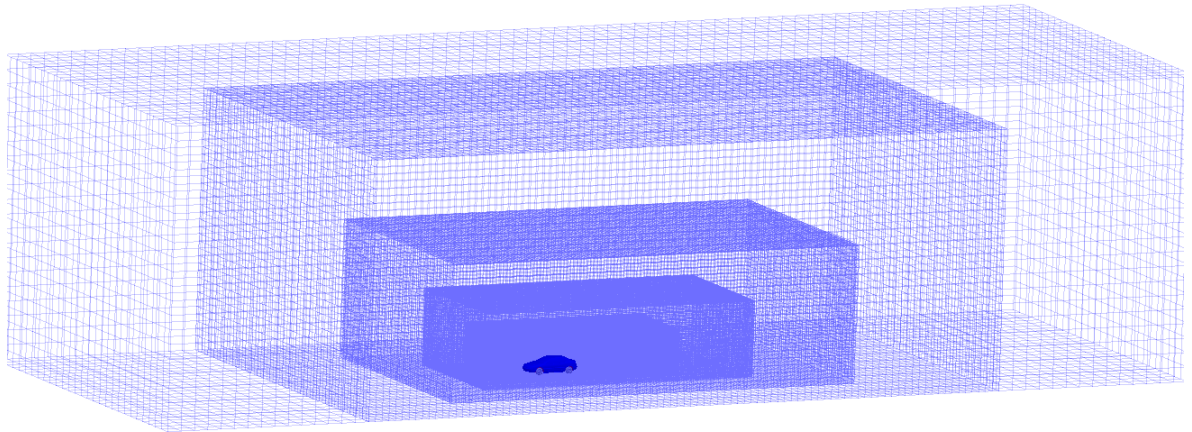


FIGURE 7.1 – Maillage utilisé autour du véhicule

La forme initiale du véhicule est montrée sur la figure 7.4. Une valeur moyenne en temps du champ direct est utilisée pour calculer l'adjoint, et les valeurs du champ direct moyen ainsi que du champ adjoint convergé sont ensuite utilisées pour calculer les sensibilités de la force de traînée sur les nœuds du maillage surfacique du véhicule. La cartographie des sensibilités est présentée sur la figure 7.2, avant l'étape de lissage. Nous pouvons remarquer que les sensibilités sont très bruitées, comme prévu.

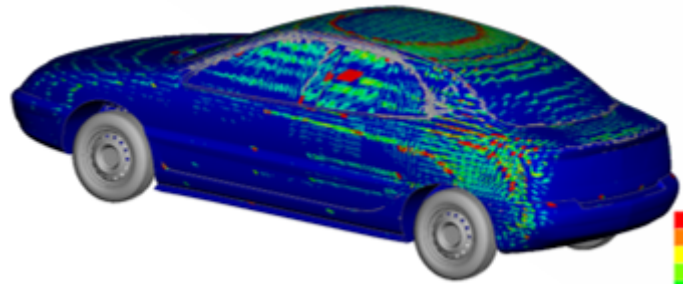


FIGURE 7.2 – Sensibilités bruitées sur le véhicule

Les cartographies d'amplitude des sensibilités après lissage sont présentées sur la figure 7.3. Les zones de sensibilités élevées sont situées sur le toit et sur l'arrière du véhicule, probablement parce qu'aplatir ces zones permettrait de rendre la voiture plus profilée et donc plus performante pour l'aérodynamique.

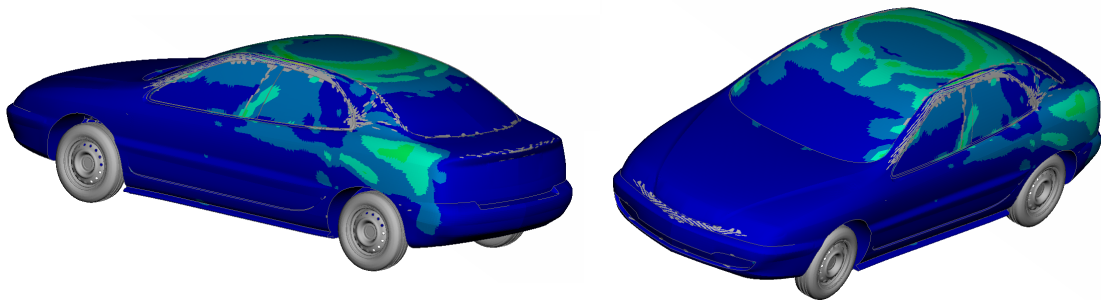


FIGURE 7.3 – Amplitude des sensibilités (les zones vertes et bleues correspondent respectivement aux zones de sensibilités fortes et faibles)

Après avoir lissé les sensibilités, le maillage du véhicule est déformé grâce à une méthode de descente à pas constant. Après deux cycles d'optimisation, la forme optimisée du véhicule est présentée sur la figure 7.5.

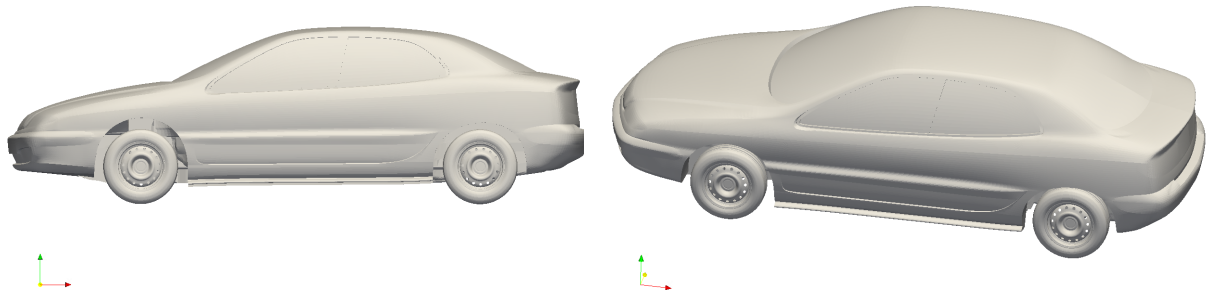


FIGURE 7.4 – Véhicule initial

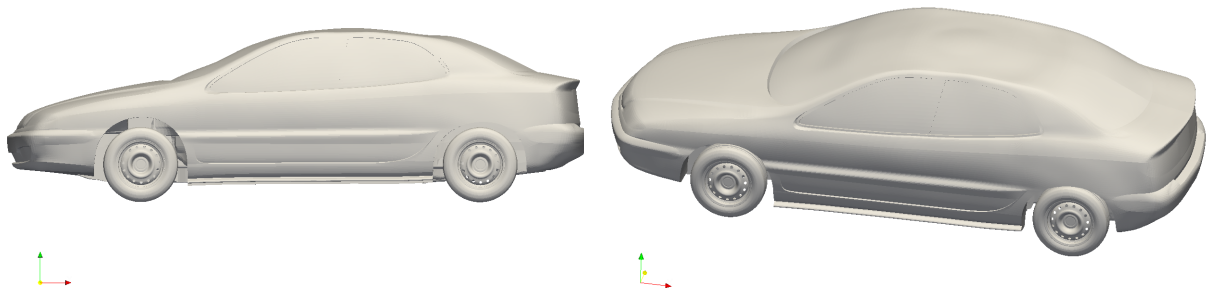


FIGURE 7.5 – Véhicule optimisé

L'algorithme d'optimisation permet de localiser les zones de sensibilités élevées, afin qu'un léger déplacement de ces zones réduise grandement la traînée. Par exemple ici, nous pouvons remarquer que l'algorithme creuse sur les bords du toit (figure 7.6), ce qui est probablement une idée non intuitive au premier abord. Le pare-brise est aussi légèrement déplacé vers l'intérieur de la voiture (figure 7.5). La méthode adjointe permet ainsi de donner des indices sur la façon de modifier la forme d'une voiture pour réduire les efforts aérodynamiques, sans simplement réduire sa surface frontale de manière globale.

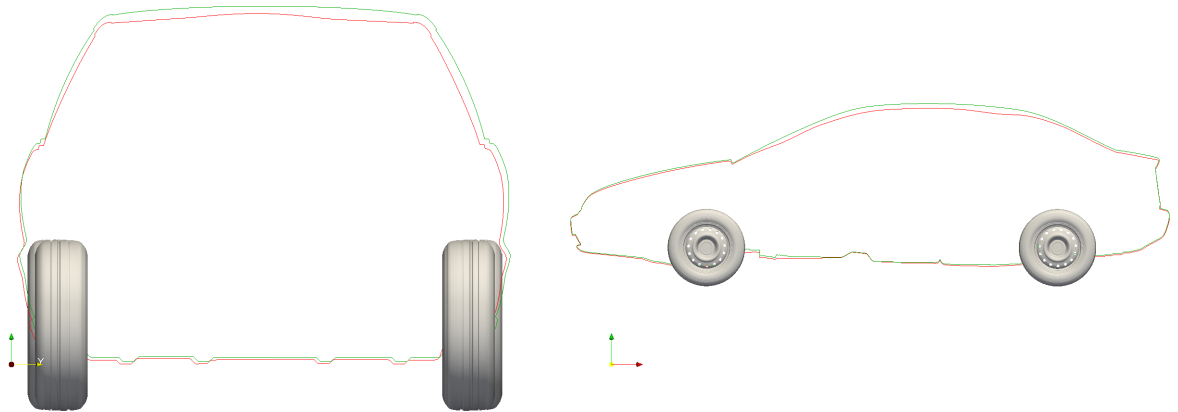


FIGURE 7.6 – Superposition de plans de coupe du véhicule initial (vert) et optimisé (rouge)

Nous pouvons aussi tracer le champ de vitesse dans un plan de coupe à l'arrière du véhicule afin de visualiser la forme du sillage pour le véhicule initial puis optimisé.

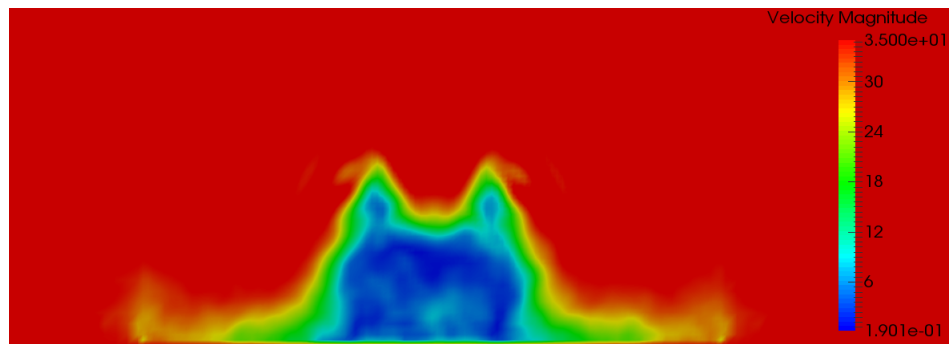


FIGURE 7.7 – Sillage derrière le véhicule initial

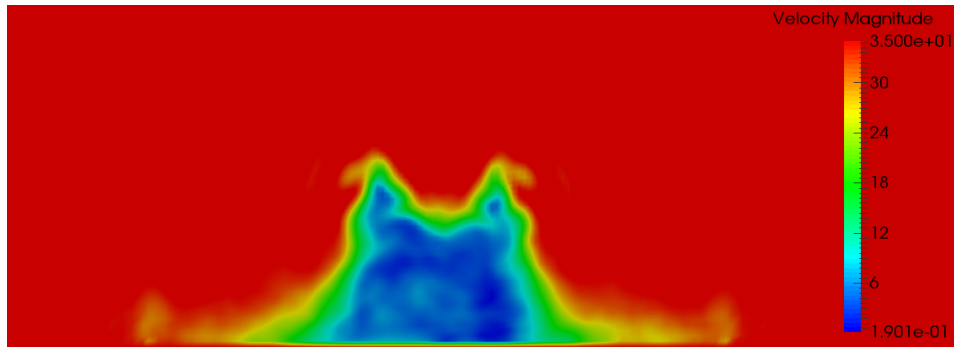


FIGURE 7.8 – Sillage derrière le véhicule optimisé

On peut remarquer que la zone bleue dans le sillage est réduite (figures 7.7 and 7.8), parce qu'il y a moins de recirculations derrière le véhicule ce qui conduit à une diminution de la force de traînée. Nous pouvons aussi remarquer que le sillage devient légèrement moins symétrique, probablement parce que la plate-forme du véhicule est asymétrique et parce que le champ direct n'est pas tout à fait symétrique non plus.

Finalement, nous pouvons aussi tracer les lignes de courant en vitesses derrière le véhicule après chaque cycle d'optimisation :

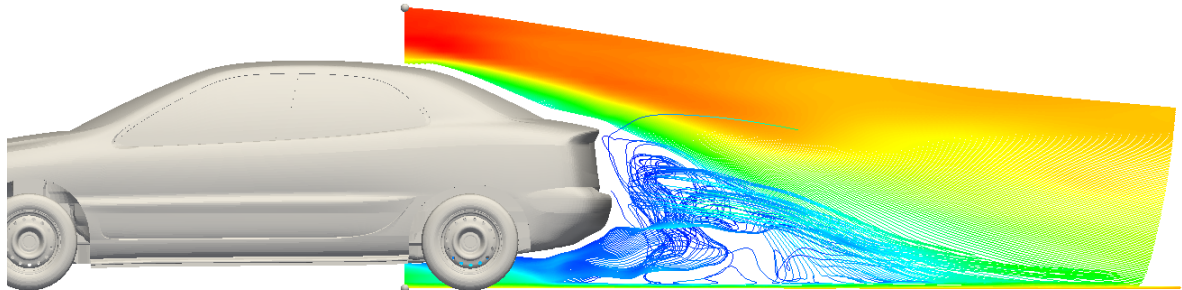


FIGURE 7.9 – Lignes de courant derrière le véhicule initial

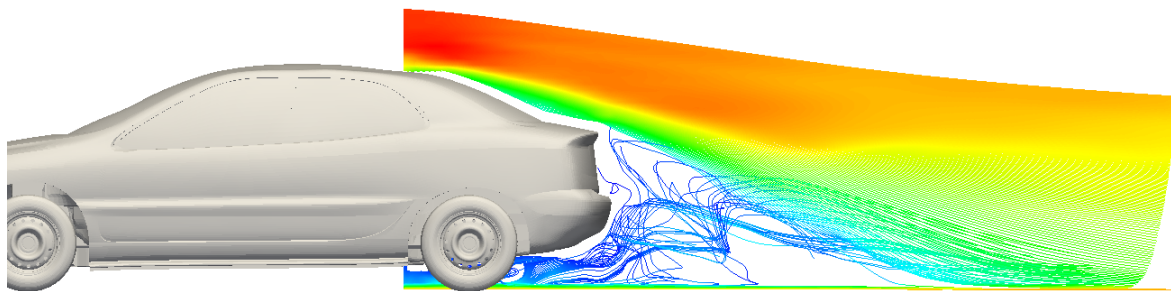


FIGURE 7.10 – cycle 1 : Lignes de courant derrière le véhicule optimisé

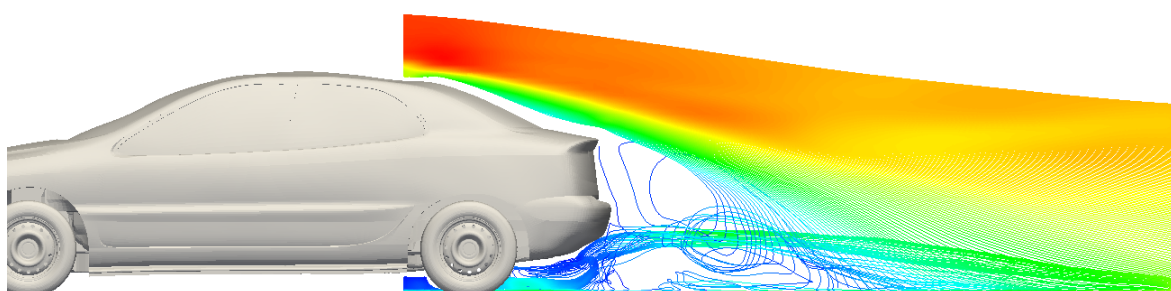


FIGURE 7.11 – cycle 2 : Lignes de courant derrière le véhicule optimisé

Nous pouvons remarquer sur les figures 7.9, 7.10 et 7.11 que les lignes de courant sont de moins en moins chaotiques derrière le véhicule, et que le décollement du sillage (la ligne de courant jaune) intervient plus en aval du véhicule. Cela est important pour la réduction des efforts aérodynamiques sur le véhicule puisque des lignes de courant qui sont moins chaotiques permettent de diminuer la force de traînée. En effet, nous avons vu dans 1.2.1 que la traînée est principalement due à la perte de pression totale entre l'amont et l'aval du véhicule, due aux phénomènes de sillage tels que la formation et maintenance de structures tourbillonnantes séparées du sillage (décollement, recirculation). Si le décollement du sillage est décalé vers l'aval du véhicule, l'intensité des tourbillons de sillage diminue dans le plan de sillage sur lequel on calcule la traînée. Cela entraîne alors la diminution de la force de traînée comme le montre la figure suivante :

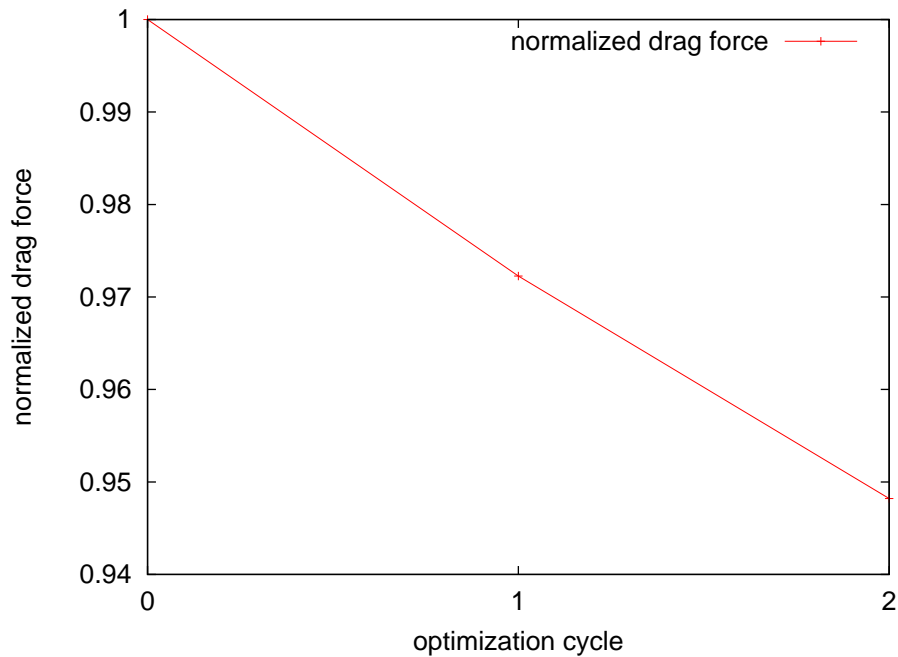


FIGURE 7.12 – Réduction de la force de traînée sur le véhicule

Sachant que l’optimisation d’un véhicule se fait nécessairement sous contraintes (sur le design par exemple), ces contraintes agissent comme critère d’arrêt de l’optimisation dans une utilisation industrielle du solveur adjoint. L’optimisation est donc arrêtée ici après deux cycles d’optimisation parce que nous avons montré que la méthode adjointe est efficace pour réduire la traînée d’au moins 5% (figure 7.12) à un coût de calcul très intéressant.

7.1 Conclusion

Le solveur adjoint a proposé avec succès des idées innovantes pour optimiser la forme de la voiture, telle que la création de sillons sur les bords du toit, ainsi qu’un pare-brise légèrement déplacé vers l’intérieur de la voiture, qui ne seraient probablement pas imaginées naturellement. Nous pourrions continuer l’optimisation pour plus que deux cycles, mais nous rappelons ici que ce cas a été utilisé pour valider la méthode. Dans les utilisations prochaines du solveur en industrie, les contraintes sur le design du véhicule pourraient servir de critère d’arrêt de l’optimisation puisque certaines parties de la voiture ne sont pas modifiables pour respecter les critères du design. De cette façon, la fonction coût (ici la force de traînée) peut atteindre un plateau une fois que le véhicule a été déformé de façon maximale en respectant les critères design.

Conclusion de la partie III

En conclusion, la méthode adjointe en LBM a été développée et validée sur un cas industriel, avec un écoulement très turbulent. Une mise en données similaire à celle utilisée en simulation aérodynamique chez Renault a été utilisée. Quelques différences sont cependant à noter : le maillage est déraffiné dans notre cas afin d'avoir des temps de calcul raisonnables, le calcul de l'écoulement sous le capot n'est pas réalisé et les roues sont fixes. Cela permet de simplifier la mise en données dans une démarche de développement de code.

Enfin, le dernier chapitre a montré l'efficacité de la méthode d'optimisation développée ici sur un cas industriel étudié chez Renault. La parallélisation du solveur a été nécessaire afin d'obtenir des temps de calcul satisfaisants pour un calcul à grandes échelles. Une étape de lissage des sensibilités grâce à un algorithme du Laplacien itéré a aussi été utilisée dans la boucle d'optimisation afin d'arriver à des résultats exploitables pour un cas industriel. Un morphing direct, nœud par nœud, a ensuite été utilisé, en conjonction avec une méthode de descente à pas constant, afin d'optimiser la forme du véhicule. La force de traînée a ainsi pu être réduite de 5% sur un véhicule à une vitesse de 130 km/h.

Conclusion générale

Après un début de thèse consacré à l'optimisation en aérodynamique interne de conduits (réduction de la perte de charge dans un conduit de climatisation par exemple), il a été montré que des outils tels que Fluent ou Star-CCM+ possèdent déjà un solveur adjoint fonctionnel qui ont donné de bons résultats sur des cas 3D utilisés chez Renault (ces études ne sont pas montrées pour des raisons de confidentialité industrielle). Ces outils reposent sur la solution adjointe des équations de Navier-Stokes. Cette première étude a ainsi permis de monter en compétences sur la méthode adjointe et d'orienter le sujet de thèse sur la réduction du coefficient de traînée d'un véhicule, dont l'aérodynamique externe est simulée avec les équations de Lattice-Boltzmann. Ce coefficient influe sur un certain nombre de paramètres dans l'industrie automobile, et surtout sur la consommation en carburant à vitesse élevée. Cette thèse avait donc pour objectif final le développement d'un solveur adjoint appliqué à la méthode de Lattice-Boltzmann.

Suite à une phase bibliographique qui a permis de passer en revue les différentes méthodes d'optimisations existantes, l'optimisation de forme surfacique par solveur adjoint a été choisie comme la méthode la plus prometteuse dans nos cas d'études. En effet, l'optimisation surfacique permet de garder la forme globale du véhicule tout en imposant de petits déplacements bien choisis qui peuvent contribuer significativement à réduire les efforts aérodynamiques. Pour déterminer où et comment effectuer ces déplacements, la méthode adjointe a été choisie car elle est la plus adaptée aux problèmes à grandes dimensions. Cette méthode permet en effet de calculer les gradients indépendamment du nombre de paramètres d'optimisation, qui sont ici les nœuds du maillage surfacique du véhicule (près de 100 000 dans le cas industriel présenté dans la dernière partie). De plus ce sujet semblait être le plus prometteur puisqu'il n'existe pas de solveur adjoint industriel en LBM à la connaissance de l'auteur.

L'aérodynamique externe étant simulée avec les équations de Lattice-Boltzmann dans le logiciel ProLB chez Renault, il a été décidé de développer le solveur adjoint dans ce code. Nous nous sommes donc appuyés sur le code existant pour simuler le champ direct, puis nous avons développé le solveur adjoint par étapes, de complexité croissante. Il a d'abord fallu poser les équations du problème direct, en cœur de fluide (modèle de collision), proche paroi (algorithme de paroi), mais aussi au niveau des raffinements de maillage (algorithmes de transition de résolution) et des conditions limites (en entrée et sortie du domaine de simulation). Chacune de ces équations a ensuite été dérivée afin d'arriver aux équations adjointes qui ont été codées dans le solveur adjoint (de la même façon en cœur de fluide, proche paroi, au niveau des raffinements de maillage, et des conditions limites). Pour cela, une approche mixte entre approche continue et approche discrète a été choisie, afin de combiner les avantages des deux méthodes. En effet nous avons dérivé les équations déjà discrétisées en espace, temps et vitesse de l'équation de Boltzmann de façon astucieuse afin d'arriver aux équations adjointes (comme dans l'approche discrète). Cela a permis d'obtenir un ensemble d'équations distinct pour le problème adjoint, que nous pouvons résoudre en utilisant les mêmes schémas numériques que pour le problème direct (comme dans l'approche continue).

Le développement par étapes du solveur adjoint nous a permis de cerner les difficultés existantes, et de proposer des solutions adéquates. Nous avons aussi remarqué que le calcul de la force de traînée dans le sillage et non sur la peau du véhicule est plus adapté dans notre cas. Ensuite, nous avons étudié les approximations temporelles possibles sur le calcul direct, et nous en avons conclu qu'utiliser une valeur moyenne du champ direct permet d'obtenir un bon compromis entre précision des gradients et coût de calcul. Finalement, nous avons utilisé les algorithmes de transitions de maillage disponibles dans ProLB pour les dériver afin de développer les algorithmes adjoints pour les transitions de maillage, et nous les avons validés sur un cas simple.

La dernière partie a montré les difficultés rencontrées lors du passage d'un cas d'étude 2D laminaire à un cas 3D turbulent à grandes échelles, à savoir que les algorithmes utilisés dans les cas 2D laminaires menaient à la divergence des cas 3D turbulents.

La méthode adjointe a ensuite été appliquée à un cas industriel utilisé chez Renault, afin de calculer les gradients surfaciques sur le véhicule. Ces gradients étant trop bruités pour être utilisés tels quels, une étape de lissage par un algorithme du Laplacien itéré a été utilisée afin d'obtenir des gradients exploitables. La dernière étape de la boucle d'optimisation consiste à utiliser un outil de morphing direct (basé sur une méthode de descente du gradient à pas constant) afin de déplacer les nœuds du maillage surfacique du véhicule. La nouvelle forme obtenue a permis de réduire la force de traînée de 5%, ce qui a montré l'efficacité du solveur adjoint dans un cas industriel turbulent.

En résumé, il est important de remarquer que les hypothèses simplificatrices retenues (telles que l'utilisation d'une valeur moyenne en temps du champ direct pour calculer un adjoint stationnaire par exemple) ont été documentées de façon rigoureuse afin d'arriver à une optimisation efficace et valide sur un cas industriel. L'étude sur des cas à complexité croissante a permis de cibler les difficultés et de proposer des solutions efficaces, à chaque étape du développement. Le solveur adjoint développé au cours de cette thèse, qui est une nouveauté au niveau industriel en LBM, a donc montré de bons résultats sur un cas industriel, et pourra être utilisé en projet chez Renault.

Perspectives

Nous pouvons imaginer plusieurs suites à ces travaux de thèse. Tout d'abord, il serait intéressant d'inclure des contraintes dans le processus d'optimisation. Ces contraintes, qu'elles correspondent à des égalités ou des inégalités, pourraient être incluses dans le Lagrangien, sachant que les inégalités seront probablement plus difficiles à résoudre que les égalités.

Afin d'être inclus dans un processus de développement industriel, il serait aussi intéressant d'obtenir les sensibilités de la force de traînée par rapport aux paramètres CAO

et non au maillage surfacique du véhicule [24]. En effet les paramètres CAO sont ceux utilisés pour fabriquer les pièces du véhicule en usine, alors que le maillage surfacique est utile en simulation numérique pour réaliser des calculs sur le véhicule.

Une autre perspective intéressante serait de réaliser une optimisation multi-objectif, comme cela a été fait récemment dans le cadre d'applications automobiles [73, 74].

Enfin, une méthode de descente plus complexe que celle du gradient à pas constant pourrait être utilisée, afin d'obtenir un pas de descente du gradient optimal.

Bibliographie

- [1] *ANSYS Fluent Adjoint Solver*, 2013.
- [2] G. Allaire, C. Dapogny, and P. Frey. Shape optimization with a level set based mesh evolution method. *Comput. Methods Appl. Mech. Engrg.*, 282 :22–53, 2014.
- [3] G. Allaire, F. Jouve, and A.M. Toader. A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334 :1125–1130, 2002.
- [4] G. Allaire, F. Jouve, and A.M. Toader. Structural optimization by the level-set method. *International Series of Numerical Mathematics*, 147 :1–15, 2003.
- [5] W. K. Anderson, J. C. Newman, D. L. Whitfield, and E. J. Nielsen. Sensitivity analysis for the Navier-Stokes equations on unstructured meshes using complex variables. *AIAA Paper*, 99 :3294, 1999.
- [6] W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA Journal*, 1997.
- [7] M. Bendsoe and N. Kichuchi. Generating optimal topologies in structural design using a homogenization method. *Computers Methods in Applied Mechanics and Engineering*, 71 :197–224, 1988.
- [8] A. Betz. Ein verfahren zur direkten ermittlung des profilwiderstandes. In *Zeitschrift fur Flugtechnik und Motorluftschifahrt (ZFM)*, 1925.
- [9] P. Bhatnagar, E. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude process in charged and neutral one-component systems. *Physcial Review E*, 94 :511–525, 1954.
- [10] V.G. Boltyanskii, R.V. Gamkrelidze, and L.S. Pontryagin. Towards a theory of optimal processes. *Dokl. Akad. Nauk SSSR*, 110 :7–10, 1956.
- [11] T. Borrvall and J. Peterson. Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids*, 41 :77 – 107, 01 2003.
- [12] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11) :3452–3459, 2001.
- [13] V. Challis and J. Guest. Level-set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering*, 79 :1284–1308, 2009.
- [14] D.D Chao and C.P. van Dam. Airfoil drag prediction and decomposition. *Journal of Aircraft*, 36 :675–681, 1999.

- [15] C. Chen, K. Yaji, T. Yamada, K. Izui, and S. Nishiwaki. Local-in-time adjoint-based topology optimization of unsteady fluid flows using the lattice Boltzmann method. *Mechanical Engineering Journal*, 2017.
- [16] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *J. Comput. Phys*, 262 :358–378, 2014.
- [17] C. Dapogny, P. Frey, F. Omnès, and Y. Privat. Geometrical shape optimization in fluid mechanics using freefem++. *Struct. Multidiscip. Optim*, 58 :2761–2788, 2018.
- [18] J. Delery. *Traité d’aérodynamique compressible : Volume 1, notions fondamentales d’aérodynamique*. 2008.
- [19] D. D’Humières. Generalized lattice Boltzmann equations, Rarefied gas dynamics : Theory and simulations. *Progress in Astronautics and Aeronautics*, 159 :450–458, 1992.
- [20] CB. Dilgen, SB. Dilgen, D. Fuhrman, O. Sigmund, and B. Lazarov. Topology optimization of turbulent flows. *Comput. Methods Appl. Mech. Engrg*, 331 :363–393, 2018.
- [21] F. Dugast, Y. Favennec, C. Josset, Y. Fan, and L. Luo. Topology optimization of thermal fluid flows with an adjoint Lattice Boltzmann Method. *Journal of Computational Physics*, 365 :376–404, 2018.
- [22] A. Dupuis and B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Physical Review E*, 67, 2003.
- [23] O. Filippova and D. Hanel. Grid refinement for lattice-BGK models. *Journal of Computational Physics*, 147 :219–228, 1998.
- [24] P. Froment. *Optimisation de formes paramétriques en grande dimension*. PhD thesis, Ecole Centrale de Lyon & Renault, 2014.
- [25] M. Geier, A. Greiner, and J.G. Korvink. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Physical Review E*, 73 :066705–458, 2006.
- [26] M. Geier, M. Schönherr, A. Pasquali, and M. Krafczyk. The cumulant lattice Boltzmann equation in three dimensions : Theory and validation. *Computers & Mathematics with Applications*, 70 :507–547, 2006.
- [27] F. Gendre. *Développement de méthodes de Boltzmann sur réseau en maillages non-uniformes pour l’aéroacoustique automobile*. PhD thesis, Laboratoire M2P2 & Renault, 2018.
- [28] K. Giannakoglou, D. Papadimitriou, D. E. Papoutsis-Kiachagias, and C. Othmer. Adjoint methods in CFD-Based Optimization - gradient computation & beyond.
- [29] I. Ginzburg and D. d’Humières. Local second-order boundary methods for lattice Boltzmann models. *J. Stat. Phys.*, 84 :927–971, 1996.
- [30] I. Ginzburg, F. Verhaeghe, and D. d’Humières. Two-Relaxation-Time lattice Boltzmann scheme : About parametrization, velocity, pressure and mixed boundary conditions. *Communications in Computational Physics*, 2007.

- [31] Z. L. Guo, C. G. Zheng, and B. C. Shi. Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Physics*, 11(4) :366–374, 2002.
- [32] I. Halliday, L.A. Hammond, and C.M. Care. Enhanced closure scheme for lattice Boltzmann equation hydrodynamics. *J. Phys. A : Mathematical and General*, 35 :157–166, 2002.
- [33] R.L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Comput. Math. Applic.*, 19 :163–208, 1990.
- [34] X. He, Q. Zou, L.-S. Luo, and M. Dembo. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. *J. Stat. Phys.*, 87 :115–135, 1997.
- [35] M. Hekmat and M. Mirzaei. Development of discrete adjoint approach based on the lattice Boltzmann method. *Advances in Mechanical Engineering*, 6, 2014.
- [36] E. Helgason. *Development of adjoint-based optimization methods for ducted flows in vehicles*. PhD thesis, Chalmers University of Technology, 2015.
- [37] A. Hopf, A. Hitchings, and L. Routledge. CFD topological optimization of a car water-pump inlet using TOSCA fluid and STAR-CCM+. In *STAR Global Conference, Vienna*, 2014.
- [38] W.-H Hucho and G Sovran. Aerodynamics of road vehicles. *Annual Review of Fluid Mechanics*, 25 :485–537, 2003.
- [39] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3) :233–260, 1988.
- [40] G. Karpouzas and E. De Villiers. Level-set based topology optimization using the continuous adjoint. In *OPT-i : An International Conference on Engineering and Applied Sciences Optimization, Greece*, 2014.
- [41] G.K. Karpouzas, E.M. Papoutsis-Kiachagias, T. Schumacher, E. de Villiers, K.C. Giannakoglou, and C. Othmer. Adjoint optimization for vehicle external aerodynamics. *International Journal of Automotive Engineering*, 7(1) :1–7, 2016.
- [42] I. S. Kavvadias, G. Karpouzas, E. Papoutsis-Kiachagias, D. Papadimitriou, and K. Giannakoglou. Optimal flow control and topology optimization using the continuous adjoint method in unsteady flows. In *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, pages 159–173, 11 2015.
- [43] M.J. Krause, B. Förster, A. Mink, and H. Nirschl. Towards solving fluid flow domain identification problems with adjoint lattice Boltzmann methods. *High Performance Computing in Science and Engineering*, 2016.
- [44] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalized shape optimization of fluids with the lattice Boltzmann method. *International Journal for Numerical Methods in Fluids*, 65 :496 – 519, 2009.

- [45] L. Laniewski-Wollk and J. Rokicki. Adjoint lattice Boltzmann for topology optimization on multi-gpu architecture. *Computers & Mathematics with Applications*, 71 :833–848, 2016.
- [46] J. Latt and B. Chopard. Lattice-Boltzmann method with regularized pre-collision distribution functions. *Mathematics and Computers in Simulation*, 72 :165–168, 2006.
- [47] J. Le-Bouette. Optimisation numérique en aérodynamique GMP. Technical report, Renault, 2016.
- [48] E. Leveque, F. Toschi, L. Shao, and J.P. Bertoglio. Shear-Improved Smagorinsky Model for Large-Eddy Simulation of Wall-Bounded turbulent flows. *Journal of Fluid Mechanics*, 570 :491 – 502, 2007.
- [49] X. Li, L. Fang, and Y. Peng. Airfoil design optimization based on lattice Boltzmann method and adjoint approach. *Applied Mathematics and Mechanics*, 39 :1–14, 2018.
- [50] C.L. Lin and Y.G. Lai. Lattice Boltzmann method on composite grids. *Physical Review E*, 62 :2219–2225, 2000.
- [51] J.L. Lions. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. *Dunod Gauthier-Villars*, 1968.
- [52] G. Liu, M. Geier, Z. Liu, M. Krafczyk, and T. Chen. Discrete adjoint sensitivity analysis for fluid flow topology optimization based on the generalized lattice Boltzmann method. *Computers and Mathematics with Applications*, 2014.
- [53] J.N. Lyness and C. B. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4 :202–210, 1967.
- [54] D. Makhija, G. Pingen, R. Yang, and K. Maute. Topology optimization of multi-component flows using a multi-relaxation time lattice Boltzmann method. *Computers and Fluids*, 2012.
- [55] P. Meliga. Computing the sensitivity of drag and lift in flow past a circular cylinder : Time-stepping versus Self-consistent analysis. *Physical Review Fluids*, 2 :073905, 2017.
- [56] H. Mitchell and N. Mashkit. Noise smoothing by a fast K-nearest neighbor algorithm. *Signal Processing-Image Communication*, 4 :227–232, 06 1992.
- [57] B. Mohammadi. Global optimization, level set dynamics, incomplete sensitivity and regularity control. *International Journal of Computational Fluid Dynamics*, 21 :61–68, 2007.
- [58] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annu. Rev. Fluid Mech*, 2325 :11–1, 01 2003.
- [59] O. Moos, F.R. Klimetzek, and R. Rossmann. Bionic optimization of air-guiding systems. *SAE 2004-01-1377*, 2004.
- [60] S. Nadarajah. *The discrete adjoint approach to aerodynamic shape optimization*. PhD thesis, Stanford University, 2003.

- [61] S.K. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. 38th Aerospace Sciences Meeting and Exhibit, Reno,NV,U.S.A., AIAA, 2000.
- [62] F. Noca, D. Shiels, and D. Jeon. Measuring instantaneous fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of Fluids and Structures*, 11 :345–350, 1997.
- [63] F. Noca, D. Shiels, and D. Jeon. A comparison of methods for evaluating time-dependent fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of Fluids and Structures*, 13 :551–578, 1999.
- [64] S. Norgaard, O. Sigmund, and B. Lazarov. Topology optimization of unsteady flow problems using the lattice Boltzmann method. *Journal of Computational Physics*, 2015.
- [65] M. Onorato, A.F. Costelli, and A. Garrone. Drag measurement through wake analysis. In *SAE Technical Paper*. SAE International, 1984.
- [66] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79 :12–49, 1988.
- [67] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *Int. J.Num.Meth.Fluids*, 2008.
- [68] C. Othmer. Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry*, 4(1) :6, 2014.
- [69] C. Othmer, T. Kaminski, and R. Giering. Computation of topological sensitivities in fluid dynamics : cost function versatility. In *European Conference on Computational Fluid Dynamics ECCOMAS CFD*, 2006.
- [70] O. Othmer, D-E. Manosalvas-Kjono, A. Jameson, and J. Alonso. Aerodynamic topology optimization : Some observations on hysteresis in separated flows. In *23rd AIAA Computational Fluid Dynamics Conference. Denver, Colorado.*, 2017.
- [71] E. Papoutsis-Kiachagias, E. Kontoleontos, A. Zymaris, D. Papadimitriou, and K. Giannakoglou. Constrained topology optimization for laminar and turbulent flows, including heat transfer. *Evolutionary and deterministic methods for design*, 2011.
- [72] E. Papoutsis-Kiachagias and K. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization : Industrial applications. *Int. J.Num.Meth.Fluids*, 2013.
- [73] E. M. Papoutsis-Kiachagias, V. G. Asouti, K. C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura. Multi-point aerodynamic shape optimization of cars based on continuous adjoint. *Struct. Multidiscip. Optim*, 59 :675–694, 2019.
- [74] G. Pierrot, J. Papper, T. Han, and S. Kaushik. Combined drag and cooling optimization of a car vehicle with an adjoint-based approach. 04 2018.

- [75] G. Pingon, A. Evgrafov, and K. Maute. A parallel Schur complement solver for the solution of the adjoint steady-state lattice Boltzmann equations : application to design optimisation. *International Journal of Computational Fluid Dynamics*, 22(7) :457–464, 2008.
- [76] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59 :117 – 128, 06 1973.
- [77] Y. H. Qian, D. D’Humières, and P. Lallemand. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Letters)*, 17(6) :479, 1992.
- [78] B.N. Rajani, A. Kandasamy, and S. Majumdar. Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modeling*, 33 :1228–1247, 2008.
- [79] D. Ricot. *Simulation numérique d’un écoulement affleurant une cavité par la méthode Boltzmann sur réseau et application au toit ouvrant de véhicules automobiles*. PhD thesis, Ecole Centrale de Lyon & Renault, 2002.
- [80] D. Ricot, V. Maillard, and C. Bailly. Numerical simulation of unsteady cavity flow using lattice Boltzmann method. *AIAA Paper*, 02 :2532, 2002.
- [81] M. Roumeas, P. Gillieron, and A. Kourta. Drag reduction by flow separation control on a car after body. *International Journal for Numerical Methods in Fluids*, 60 :1222 – 1240, 2009.
- [82] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, 1994.
- [83] P. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E*, 48 :4823–4842, 1993.
- [84] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM*, 40 :110–112, 1998.
- [85] M.M Tekitek, M. Bouzidi, F. Dubois, and P. Lallemand. Adjoint lattice Boltzmann equation for parameter identification. *Computers and Fluids*, 2005.
- [86] H. Toubin. *Prediction and Phenomenological Breakdown of Drag for Unsteady Flows*. PhD thesis, ONERA, 2015.
- [87] D. Touil, D. Ricot, and E. Lévêque. Direct and Large-Eddy Simulation of turbulent flows on composite multi-resolution grids by the lattice Boltzmann method. *Journal of Comput. Physics*, 256 :220–233, 2013.
- [88] C.P. van Dam. Recent experience with different methods of drag prediction. *Progress in Aerospace Sciences*, 35 :751–798, 1999.
- [89] E. Vergnault and P. Sagaut. Application of lattice Boltzmann method to sensitivity analysis via complex differentiation. *Journal of Computational Physics*, 2011.
- [90] E. Vergnault and P. Sagaut. An adjoint-based lattice Boltzmann method for noise control problems. *Journal of Computational Physics*, 2014.
- [91] J. Verschaeve and B. Muller. A curved no-slip boundary condition for the lattice Boltzmann method. *J. Comp. Phys.*, 229 :6781–6803, 2010.

- [92] S. Wilhelm, J. Jacob, and P. Sagaut. An explicit power-law-based wall model for lattice Boltzmann method-Reynolds-averaged numerical simulations of the flow around airfoils. *Physics of Fluids*, 30 :065111, 2018.
- [93] S. Wolfram. Cellular automata fluids 1 : Basic theory. *J. Stat. Phys.*, 45 :471–526, 1986.
- [94] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K. Izui, and S. Nishiwaki. Topology optimization in thermal-fluid flow using the lattice Boltzmann method. *J. Comp. Phys.*, 2015.
- [95] Q. Zhou, T. Weinkauff, and O. Sorkine-Hornung. Feature-based mesh editing. In *Eurographics*, 2011.