



**HAL**  
open science

# Adaptive residual based schemes for solving the penalized Navier Stokes equations with moving bodies : application to ice shedding trajectories

Léo Nouveau

► **To cite this version:**

Léo Nouveau. Adaptive residual based schemes for solving the penalized Navier Stokes equations with moving bodies : application to ice shedding trajectories. General Mathematics [math.GM]. Université de Bordeaux, 2016. English. NNT : 2016BORD0410 . tel-01500093

**HAL Id: tel-01500093**

**<https://theses.hal.science/tel-01500093>**

Submitted on 2 Apr 2017

**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.

THÈSE  
PRÉSENTÉE À  
L'UNIVERSITÉ DE BORDEAUX  
ÉCOLE DOCTORALE DE MATHÉMATIQUES ET  
D'INFORMATIQUE  
par **Léo Nouveau**  
POUR OBTENIR LE GRADE DE  
**DOCTEUR**  
SPÉCIALITÉ : MATHÉMATIQUES APPLIQUÉES

---

**Adaptive Residual Based Schemes For Solving The  
Penalized Navier Stokes Equations With Moving  
Bodies.  
Application To Ice Shedding Trajectories.**

---

**Directeurs de thèse :**

Mario RICCHIUTO    Directeur de Recherche, INRIA  
Héloïse BEAUGENDRE    Maître de Conférence (HDR), INRIA  
Cécile DOBRZYNSKI    Maître de Conférence, INRIA

**Rapporteurs :**

Guglielmo SCOVAZZI    Professeur, Duke University  
Philippe VILLEDIEU    Professeur, ONERA

**Date de soutenance :**    16 Décembre 2016

**Devant la commission d'examen composée de :**

Rémi ABGRALL	.....	Professeur, Universität Zürich	.....	Président du jury
Héloïse BEAUGENDRE		Maître de Conférence (HDR), INRIA		Directrice de Thèse
Hervé GUILLARD	....	Directeur de Recherche, INRIA	.....	Examineur
Mario RICCHIUTO	...	Directeur de Recherche, INRIA	.....	Directeur de Thèse
Philippe VILLEDIEU	.	Professeur, ONERA	.....	Rapporteur



---

**Abstract** The prediction of solid motion evolving in a fluid presents a real interest for engineering application such as ice accretion on aerodynamics bodies. In this context, considering de-icing systems, the ice shedding trajectory is needed to prevent the risk of collision/ingestion of the ice in/with some sensitive part of the aircraft. This application raises many challenges from a numerical point of view, especially concerning mesh generation/adaptation as the solid moves in the computational domain. To handle this issue, in this work the solids are known implicitly on the mesh via a level set function. An immersed boundary method, called penalization, is employed to impose the wall boundary conditions. To improve the resolution of these boundaries, the equations are solved on adaptive unstructured grids. This allows to have a refinement close to the solid boundary and thus increases the solid definition, leading to a more accurate imposition of the wall conditions. To save computational time, and avoid costly remeshing/interpolation steps, the strategy chosen for unsteady simulations is to use a constant connectivity mesh adaptation, also known as r-adaptation

**Résumé** La prédiction de mouvement de solide évoluant dans un fluide présente un réel intérêt pour des applications industrielles telle que l'accrétion de glace sur des surfaces aérodynamiques. Dans ce contexte, en considérant des systèmes de dégivrage, la prévision des trajectoire de glace est nécessaire pour éviter des risques de collision/ingestion de glace sur/dans des zones sensibles de l'avion. Ce type d'application soulève de nombreux challenges d'un point de vue numérique, en particulier concernant la génération/l'adaptation de maillage au cours du mouvement du solide dans le domaine. Pour gérer ces difficultés, dans cette étude, les solides sont définis de manière implicite via une fonction level set. Une méthode de type frontière immergée, appelée Pénalisation, est utilisée pour imposer les conditions de bords. Pour améliorer la précision de l'interface, les équations sont résolues sur des maillages non structurés adaptatifs. Cela permet d'obtenir un raffinement proche des bords du solide et ainsi d'améliorer sa définition, permettant un meilleure imposition des conditions de bord. Pour économiser du temps de calcul, et éviter de coûteuses étapes de remaillage/interpolation, la stratégie adoptée pour les simulations instationnaires est d'utiliser une adaptation de maillage à connectivité constante, aussi appelée r-adaptation.

**Keywords** Penalization, Moving Bodies, Fluid Structure Interaction, Residual Distribution Schemes, Mesh Adaptation

**Mots-clés** Pénalisation, Objets Mobiles, Interaction Fluide Structure, Schémas aux résidus distribués, Adaptation de maillage

**Laboratoire d'accueil** INRIA Bordeaux Sud Ouest - Equipe CARDAMOM, 200 avenue de la vieille tour, 33405 Talence Cedex

---

# Acknowledgement

Cette thèse a été possible et s'est déroulée dans d'excellentes conditions grâce à de nombreuses personnes que je tiens à remercier.

L'encadrement étant l'un des facteurs clé pour le bon déroulement d'une thèse, mes remerciements vont en tout premier lieu aux personnes qui m'ont encadré, apportés conseils et soutien durant ce projet.

Ce travail a été possible grâce à Héloïse et Cécile, qui après m'avoir pris en stage, m'ont permis de continuer ces travaux à leur côté, en me proposant cette thèse. Je les remercie énormément pour leur implication et le temps qu'elles m'ont accordés. La porte de leur bureau m'a toujours été ouverte et j'ai particulièrement apprécié leur disponibilité pour m'apporter aides et conseils. Outre le côté professionnel, je tiens également à les remercier pour leur attention et leur soucis de ce que cette thèse m'apporterait, me remotivant chaque fois que nécessaire. Pour toutes ces raisons, je tiens à leur dire un grand merci.

Ensuite, mes remerciements vont s'adresser à Mario, qui a accepté de s'adjoindre à la direction de ma thèse en cours de route. Son expertise et ses connaissances (qui m'impressionneront toujours!) ont été d'une aide précieuse. Il a toujours accepté de trouver le temps nécessaire pour répondre à mes interrogations, y compris pendant ses week ends ou ses vacances. Je resterais également impressionner par sa technique d'écriture au tableau qui prend très rapidement un caractère plus qu'aléatoire! Pour son implication dans mes travaux, pour son investissement dans mes projets d'après thèse, je veux l'assurer de toute ma gratitude.

Je tiens également à remercier Rémi Abgrall qui malgré son départ à Zurich a encadré ma thèse pendant la première année, et a toujours eu des commentaires avisés par mail ou au détour d'une conférence.

Cette thèse étant financé par le projet Européen STORM, je tiens à remercier toutes les personnes impliquées et les différents partenaires, sans qui les travaux de ces trois dernières années n'auraient pu être possible.

Je voudrais également remercier les différents membres du jury pour avoir accepté d'évaluer mon travail, en particulier les deux rapporteurs, Pr. Scov-

---

azzi et Pr. Villedieu, pour leur lecture minutieuse de ce manuscrit ainsi que leurs remarques et commentaires.

Au sein d'INRIA, j'ai pu travailler dans une ambiance assez extraordinaire, et ce grâce à la contribution de nombreuses personnes.

Dans l'équipe CARDAMOM, j'ai particulièrement apprécié la bonne humeur de Nico et Anne Laure qui m'ont apporté leur aide dans le micmac administratif qu'apportait chaque mission et je les en remercie.

La bonne ambiance de notre Open Space (que je qualifierais plus de Little Italie!) a toujours été au rendez vous, notamment grâce à Andrea et Luca qui préféreraient communiquer en criant (bien qu'ils qualifient ça de parler) plutôt que de se déplacer, mais également Umbe, Nikos et Seb (qui durant les derniers mois a du subir mes joies et désespoirs quand nous étions les deux derniers présents le soir). Mais ça ne se limite pas uniquement à notre (génialissime) open space. La présence de Andrea, Ghina, François, Algiane, Mickael, Maria, Stevan, Quentin, Mathieu et les échanges que j'ai pu avoir avec Pietro ont contribué à une ambiance que j'espère sincèrement retrouvé dans mes futurs projets. De nombreux stagiaires ont également participé à la bonne humeur de l'équipe au cours de ces 3 ans, et je remerciais parmi eux tout particulièrement Antoine Fondanèche qui m'a aidé pendant son stage de fin d'études et permis d'économiser un temps qui m'a été précieux.

Je suis également extrêmement heureux des rencontres faites avec des personnes d'autres équipes avec qui j'ai passé de très bons moments (et pas nécessairement des soirées!), Charles, Chris, Jo, Lucille, Camille, Cyril, Adrien (et tous ceux que j'oublie).

Evidemment, je me dois de remercier François Rué, qui a toujours énormément apprécié (j'en suis sur!) mes visites lors de problèmes de compilation ou interrogations sur PlaFrim.

J'ai également eu la chance de pouvoir parler et décompresser (plus ou moins souvent) avec de nombreux amis qui me sont chères, les Arnaud, KBesseau, Gabby (le nombre de bières descendus pour révolutionner la recherche!), Mathilde, Marina, Sonia, Marine, Boub, Thomas (et ceux que j'oublie). Merci à vous! Un petit paragraphe spécial pour Clément, mon colloc, qui a presque réussi à me convertir à LOL, mais je serais resté stoïque face à ses tentatives. Même si pour qu'on se voit en semaine il fallait que je fasse l'effort (j'aurais peut être du mettre des guillemets) d'aller au Swing Marine, j'ai "vla kiffé" ces 2 années de collocation! Maintenant, tout ce que je peux espérer c'est que vous vous occuperez bien de Karmeliet avec Nelson et Alice le temps que je la récupère!

En dernier lieu, j'adresse d'immenses remerciements (qui sont probablement les plus importants) à ma famille, en commençant par mes parents. Sans

---

eux, je n'en serais pas là aujourd'hui, et notamment grâce au soutien de ces trois dernières années. Autant quand ça se passe bien on n'a pas grand chose à raconter, autant quand les problèmes s'accumulent, on évacue, et c'est principalement ce qu'ils ont entendus. Ils ont toujours porté un regard extérieur et objectif qui m'a permis de relativiser et destresser un peu. Evidemment je ne laisserais pas en reste ma soeur, mes grands parents, oncles, tantes, cousins et cousines qui une fois passée les blagues de savoir si à force de chercher je trouvais, ont toujours su l'importance et les enjeux de ces travaux, ainsi que la quantité de travail demandé, spécialement dans des périodes de rush, même si cela implique d'annuler sa présence au dernier moment pour les anniversaires. Encore une fois, un grand merci à eux, j'ai la chance de faire partie d'une famille formidable!

---

# Résumé Substantiel

Dans le contexte de la conception d'avions, un problème concerne leur habilité à empêcher ou limiter l'accrétion de glace et de proposer des systèmes de dégivrage ou d'antigivrage. En effet, sous certaines conditions de vol, due à la présence de gouttelettes d'eau dans l'air et de faible température, des blocs de glace peuvent se former sur certaines zones d'un avion. Outre une perte de performance aérodynamique, d'importants blocs peuvent se détacher. Ce détachement est d'une importance capitale lors de la conception d'avion étant donné les dommages que ces blocs peuvent causer. Les exemples critiques sont l'ingestion par un moteur ou la collision avec une zone sensible de l'avion. La prévision des trajectoires est donc d'une très grande importance et est l'un des sujets d'étude du projet européen STORM, dont l'objectif est d'améliorer les connaissances et outils numériques de différents aspects touchant au phénomène du givrage.

Dans cette thèse, l'intérêt est porté sur le développement d'outils numériques permettant la prévision des trajectoires des blocs de glace. Deux types de modèles sont couramment utilisés, caractérisés par leur degré de fidélité. Le premier type appelé 'low fidelity' fait l'hypothèse que le bloc de glace n'influe pas sur l'écoulement de l'air. Le deuxième type, appelé 'high fidelity' propose au contraire un couplage complet pour l'interaction fluide structure, et c'est dans ce cadre que se situe la méthode adoptée. Dans cette approche 'high fidelity', trois grandes catégories de méthodes peuvent être référencées. La première, bien que non appliquée dans la littérature au contexte spécifique du givrage, propose une discrétisation explicite de l'objet dans le maillage, et utilise des techniques de déformation de maillage et remaillage/interpolation pour déplacer l'objet. La deuxième, communément appelée *méthode chimère*, propose deux niveaux (au moins) de maillage. Le premier, plus grossier, discrétise tout le domaine. Le second, plus fin, maillonne une zone restreinte proche de l'objet, et contient une discrétisation explicite de la frontière. Cette zone raffinée se déplace avec l'objet. Des interpolations d'un maillage à l'autre sont ensuite effectuées à chaque pas de temps pour résoudre le problème. Le dernier type de méthode appelé *méthode de frontière immergée* propose de maillonne tout le domaine indépendamment de la géométrie du solide considéré, qui est alors connu implicitement *via* une fonction *level set*, et les conditions de bords sont prises en compte différemment avec un terme source imposé soit au niveau

---

discret soit au niveau continu. Les étapes de remaillage/interpolation bien que limitées dans la première approche soulèvent les problèmes de temps de calcul et/ou conservation, ce dernier point étant également retrouvé avec les méthodes chimères à toutes les itérations en temps.

Dans ces travaux, étant donné l'absence de contraintes liées à la génération de maillage pour des géométries complexes (connues implicitement), la *Pénalisation*, méthode de frontière immergée, a été employée. Pour également se soustraire aux problèmes de remaillage/interpolation, il a été choisi de réaliser les simulations sur grilles non structurées adaptatives. Cela permet en effet d'améliorer la définition de l'objet en raffinant le maillage proche de sa frontière, mais également la précision de la solution en adaptant à un paramètre physique (vorticité par exemple). Bien qu'initialement envisagée pour des simulations instationnaires avec objets mobiles, une adaptation basée sur des techniques de remaillage/interpolation a voulu être évitée. En effet, comme rappelé précédemment, une telle approche peut facilement endommager la précision de la solution, il est difficile de garantir la conservation, et une telle procédure peut se révéler très coûteuse, spécialement lors de simulations parallèles. C'est pourquoi une méthode d'adaptation de maillage à connectivité constante a été mise en place, et les équations ont été résolues dans un formalisme *Arbitrary Lagrangian Eulerian* (ALE), permettant la construction de schémas conservatifs et la prise en compte de la mobilité du maillage directement dans les équations. Les schémas employés se situent dans le cadre des *résidus distribués*, et possèdent de nombreuses analogies avec les schémas *Elements Finis Stabilisés*.

En tout premier lieu (chapitre 2), pour montrer l'aspect compétitifs des méthodes de frontières immergées sur maillage adapté avec des simulations fittées (résolution des équations de Navier Stokes 'classiques' sur maillage contenant une discrétisation explicite de l'objet), des études stationnaires ont été réalisées. Les schémas aux résidus distribués sont étendus à la Pénalisation *via* une simple discrétisation de Galerkin du terme source correspondant. Un nombre limité d'adaptation de maillages utilisant des métriques permettant le contrôle d'une erreur (approximation de la level set 0 ou erreur d'approximation d'un paramètre physique) a été utilisée pour augmenter le degré de précision de la solution. Différents exemples *2D* et *3D* prouvent la faisabilité d'une telle approche.

L'étude a ensuite portée sur la résolution d'équations d'advection diffusion (et appliqué au système Navier Stokes) sur des grilles fixes avec des schémas aux résidus distribués (chapitre 3). Ces schémas ont ensuite été appliqués au système d'équations de Navier Stokes Pénalisé (toujours sur grille fixe), le terme source de Pénalisation étant cette fois traité *via* un splitting de Strang, permettant une flexibilité dans le choix du schéma numérique et un ordre théorique de deux. Deux calculs de force ont également été proposés. L'un, spécifique au splitting, est basé sur un bilan de moment entre l'étape de réso-

---

lution des équations de Navier Stokes et l'étape de Pénalisation. Le deuxième correspond à l'interpolation depuis le maillage de calcul sur un maillage surfacique du solide de la pression et des contraintes de cisaillement pour effectuer un calcul intégral sur cette surface. L'objectif de cette thèse étant de simuler les trajectoires d'objets dont la position est régie par le deuxième principe de Newton, ce calcul des forces est un élément clef. En effet, la trajectoire est directement liée aux forces aérodynamiques. Des exemples ont permis de mettre en avant la nécessité du traitement spécifique de la reconstruction des gradients à l'interface fluide/solide. Cette théorie a été appliquée à l'étude du détachement des allées de Von Karman derrière un cylindre et des cas tests de Rayleigh. Les résultats ont permis de valider l'approche employée et les calculs de force proposées.

Le coeur du sujet, à savoir la simulation d'objets mobiles a ensuite été étudié, ainsi que la mise en place de la technique d'adaptation de maillage instationnaire à connectivité constante. La première étape a été d'étendre les schémas proposés au contexte ALE. Ils ont été validés sur des équations de conservations d'advection diffusion scalaires et sur le système de Navier Stokes.

Pour l'adaptation de maillage, en se basant sur la littérature, deux techniques ont été étudiées, toutes deux basées sur une *fonction monitrice*. La première propose d'assimiler le maillage à un matériau régi par le problème d'élasticité. Une force définie comme le gradient de la fonction monitrice est appliquée pour définir les zones de raffinement. La deuxième méthode quant à elle utilise la fonction monitrice pour équidistribuer les points. La validation du processus complet d'interaction fluide structure a été réalisé en différentes étapes. La première a été de réaliser des simulations où la vitesse du solide est connu analytiquement pour se comparer à la littérature et valider le calcul des forces dans le cadre d'objets advectés sur grille adaptative (cylindre et naca0015 oscillant). La deuxième étape a été de valider le couplage calcul des forces aérodynamiques/vitesse du solide en comparant les résultats à la littérature (cylindre tombant dans une cavité). Une fois ces résultats validant notre méthode, l'application au givrage a été mise en place avec le cas test du projet STORM correspondant au lâcher en soufflerie d'une forme de glace dénommée *GLC*. Notre méthode n'étant pas couplée à un modèle de turbulence, les simulations (2D) ont été réalisées avec un Reynolds inférieur aux expériences effectuées en soufflerie et aux résultats obtenus par les partenaires du projet STORM (ONERA et DLR). Cependant, les résultats ont montré des comportements similaires aux données expérimentales et simulations du DLR et de l'ONERA, à savoir une trajectoire quasi linéaire et une oscillation de la forme de glace. Ces résultats, bien que préliminaires, ont démontré le potentiel de l'approche proposée dans ces travaux, l'utilisation d'une méthode de frontière immergée couplée à une adaptation de maillage à connectivité constante.

---

# Contents

<b>Contents</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation	1
1.2 Moving Object Simulation and Ice Shedding Trajectory, State of the art	2
1.2.1 Low fidelity models	2
1.2.2 High fidelity models	4
1.2.2.1 Moving Mesh/Remeshing	4
1.2.2.2 Chimera methods	5
1.2.2.3 Immersed Boundary methods	5
1.3 Description of the proposed approach	9
1.4 Thesis Contributions and Manuscript organization	9
<b>2 Steady Penalized Navier Stokes Equations</b>	<b>13</b>
2.1 Problem Statement	14
2.1.1 Physical Model, Navier Stokes equations	14
2.1.1.1 Navier Stokes Equations	14
2.1.1.2 Boundary Conditions	16
2.1.2 Penalized Navier Stokes equations	16
2.1.2.1 Modification of the equations	16
2.1.2.2 Signed Distance Function	17
2.1.3 Dimensionless form of the equations	18
2.1.4 Simplified Model - Scalar Conservation Law	20
2.2 Mesh Adaptation for Steady Simulation	22
2.2.1 State of the art	22
2.2.2 Some metric Notions	23
2.2.3 Metric Expression For Mesh Adaptation	24
2.2.3.1 Principle	24
2.2.3.2 Physical Adaptation	25
2.2.3.3 Adaptation to the Signed Distance	26
2.2.3.4 Metric intersection	27
2.2.4 Algorithm	28

---

2.3	Discretization with Residual Schemes . . . . .	31
2.3.1	State of the art . . . . .	31
2.3.2	Notations for mesh and approximate solution . . . . .	32
2.3.3	Steady scalar conservation laws . . . . .	33
2.3.3.1	Principle and construction . . . . .	33
2.3.3.2	Basic Properties . . . . .	36
2.3.3.3	Analogy with Stabilized Finite Elements . . . . .	39
2.3.3.4	Two examples of RD schemes . . . . .	40
2.3.3.5	Extension to advection/diffusion problems . . . . .	41
2.3.4	Extension to the penalized Navier Stokes system . . . . .	43
2.3.4.1	RD schemes for the Navier Stokes equations . . . . .	43
2.3.4.2	Modification of the schemes for the Penalization . . . . .	44
2.3.5	Penalized Navier Stokes Results . . . . .	46
2.3.5.1	2D Naca0012 airfoil . . . . .	46
2.3.5.2	2D supersonic flow past wedge . . . . .	48
2.3.5.3	3D supersonic flow past extruded wedge . . . . .	51
2.4	Summary . . . . .	51
<b>3</b>	<b>Unsteady Penalized Navier Stokes Equations For Motionless Bodies</b>	<b>53</b>
3.1	Residual Schemes for Unsteady Problems . . . . .	54
3.1.1	State of the art . . . . .	54
3.1.2	Principle . . . . .	54
3.1.3	Basic Properties . . . . .	55
3.1.4	Mass Matrix Formulations . . . . .	57
3.1.5	Discretization of scalar advection diffusion problem . . . . .	58
3.1.5.1	Explicit Second Order Runge Kutta Scheme . . . . .	58
3.1.5.2	Implicit $\theta$ scheme . . . . .	60
3.1.5.3	Convergence Study . . . . .	63
3.1.6	Extension to Navier Stokes Equations . . . . .	64
3.1.6.1	Formulation of the mass matrix . . . . .	64
3.1.6.2	Second Order Explicit Runge Kutta Scheme . . . . .	66
3.1.6.3	Implicit $\theta$ Scheme . . . . .	66
3.1.6.4	Convergence study . . . . .	67
3.2	Extension to Penalization . . . . .	69
3.2.1	A splitting approach . . . . .	69
3.2.2	The Strang Splitting . . . . .	69
3.2.3	Application to penalization . . . . .	70
3.2.3.1	Development of the scheme . . . . .	70
3.2.3.2	Constant penalty variable . . . . .	72
3.2.3.3	Time dependant penalty variable . . . . .	73
3.2.4	Fluid Structure Interaction : Evaluation of the Aerodynamic Forces . . . . .	73

---

3.2.4.1	Change of Momentum Computation . . . . .	74
3.2.4.2	Surface Integral Computation . . . . .	74
3.2.5	Results . . . . .	75
3.2.5.1	Rayleigh test cases . . . . .	75
3.2.5.2	Flow Past Cylinder . . . . .	77
3.3	Summary . . . . .	81
<b>4</b>	<b>Penalization and Moving Bodies - An ALE Approach</b>	<b>85</b>
4.1	Residual Distribution Scheme For Solving Arbitrary Lagrangian Eulerian Problems . . . . .	86
4.1.1	State of the art . . . . .	86
4.1.2	Conservation Laws In ALE Formulation . . . . .	87
4.1.2.1	Generalities and notations . . . . .	87
4.1.2.2	ALE Integral form of a conservation law . . . . .	88
4.1.2.3	Discrete Geometric Conservation Law . . . . .	91
4.1.3	RD Schemes for scalar conservation law in ALE Form . . . . .	92
4.1.4	General Formulation . . . . .	92
4.1.4.1	Explicit RK2 Scheme . . . . .	94
4.1.4.2	Implicit $\theta$ Scheme . . . . .	95
4.1.4.3	Convergence Study . . . . .	96
4.1.5	Extension To Navier Stokes equations . . . . .	97
4.1.5.1	Formulation of the schemes . . . . .	97
4.1.5.2	Convergence Study . . . . .	98
4.2	r-Adaptation for moving bodies . . . . .	100
4.2.1	State of the art . . . . .	100
4.2.2	An Elasticity Based Model . . . . .	101
4.2.2.1	Problem Statement . . . . .	101
4.2.2.2	Finite Element Resolution . . . . .	101
4.2.2.3	Monitor function definition . . . . .	103
4.2.3	A Laplacian Based Model . . . . .	105
4.2.3.1	Problem Statement . . . . .	105
4.2.3.2	Finite Element Resolution . . . . .	107
4.2.3.3	Monitor Function Definition . . . . .	108
4.2.4	Comparison of the approaches . . . . .	110
4.3	Penalized Navier Stokes Equations with Moving Bodies . . . . .	114
4.3.1	Moving object simulation - process . . . . .	114
4.3.2	Results . . . . .	116
4.3.2.1	In-line oscillating cylinder in a flow at rest . . . . .	116
4.3.2.2	Oscillating Naca Airfoil . . . . .	117
4.3.2.3	2D Falling Cylinder . . . . .	121
4.3.2.4	GLC305-rime ice shape . . . . .	126
4.4	Summary . . . . .	130

<b>5 Conclusion</b>	<b>133</b>
<b>A Advection and Diffusion Tensors for Navier-Stokes Equations</b>	<b>137</b>
A.1 Euler Jacobian . . . . .	137
A.2 Navier Stokes Tensor . . . . .	138
<b>Bibliography</b>	<b>141</b>

# Chapter 1

## Introduction

### 1.1 Motivation

The interest in CFD has greatly increased these last decades because it may help the understanding and prediction of some physical phenomena. The range of application is tremendous with industrial, environmental and medical purposes.

In the specific context of aircraft conception, an issue concerns the ability of the planes to prevent or limit ice accretion and to propose anti/de icing systems. Indeed, under some flight conditions, especially when passing through a cloud (common situation at landing or take off), due to the presence of droplets in the air and low temperatures, an accretion of ice may appear on some parts of the plane. It may lead to dangerous situations according to the position and amount of ice. In addition to a potential loss of aerodynamical performances, ice blocks can be released (due to the use of a de icing system for instance). Ice release is of concern to aircraft manufacturers because of potential ice debris impact on aerodynamic surfaces and/or ingestion by engines. This concern is under study within the European project STORM<sup>1</sup> regrouping different industrial partners and research centers. The aim of this project is to improve the knowledge and the numerical tools concerning the whole inflight icing process : from the ice accretion to the ice shedding trajectory prediction along with an investigation on innovative ice protection systems. STORM ambition is to provide a significant step forward in ice simulation and protection systems for aircraft engines. In this PhD thesis, we are interested in the numerical tools predicting ice block trajectories. There are generally two types of model used to track shed ice pieces, which are distinguished by their level of fidelity. The first type of model (“low-fidelity”) makes the assumption that ice pieces do not significantly affect the flow field, and the second type of model (“high-fidelity”) intends to take into account ice pieces interacting with the flow.

---

<sup>1</sup>European Union Seventh Framework Programme FP7/2007–2013 under grant agreement no. 605180 <http://www.fp7-storm.eu/>

## 1.2 Moving Object Simulation and Ice Shedding Trajectory, State of the art

### 1.2.1 Low fidelity models

The main characteristic of those models is the strong assumption that the flow is not affected by the solid. To compute the considered particle trajectory, the forces acting on it are extracted from a database according to its position, orientation and flow characteristics. The particle itself has no effect on the surrounding airflow. The advantage of this type of model resides in the use of classical CFD tools. However, the main drawback is the necessity to have *a priori* a database containing aerodynamical forces and moments for each specific solid shape studied. This requires either experiments in wind tunnel (fastidious and expensive), either numerical computations, or empirical models.

Following this idea, Chandrasekharan *et. al.* [42] used a modified water droplet trajectory code to track trajectories of an ice disk and two ice debris. In this study, the trajectories of the ice pieces were assumed to be dependant only on drag. Nevertheless, for large ice fragments, lift and aerodynamic moments can have a considerable influence on the trajectory. Three, four and six degree of freedom (DOF) models have been developed to compute ice shedding trajectories in 2D and 3D flowfields being uniform or non-uniform. Kohlman *et. al.* [103] proposed a trajectory simulation method, based on a 4-DOF model, to compute the trajectories of ice particles, represented by square plates of uniform thickness, into a uniform velocity field. Lift and drag were assumed to be the main aerodynamic forces and the rotation was limited to a single axis. Lift and drag were obtained from empirical correlations. Santos *et. al.* [144] used a similar method, but the trajectories were calculated into a non-uniform flow field around a wing. The initial position and velocity were varied, and the probability of an ice collision with the aircraft surface at a location two chords downstream of the leading edge was obtained. In those previous cited models it was assumed that only drag force, lift and pitching moment act on the ice particle. Initiated in 2003, a long-term research program at Wichita State University (WSU) has been devoted to develop and validate ice shedding analysis tools based on 4-DOF and 6-DOF model. For 6-DOF model not only drag, lift and pitching moment act on the ice particle but also the side force, rolling moment and yawing moment. Figure 1.1 shows a representation of the force  $F$  (resultant of the normal, axial and side forces) and the moment  $M$  (resultant of the rolling  $M_{bx}$ , pitching  $M_{by}$  and yawing  $M_{bz}$  moments) acting on the ice fragment. More recently, Papadakis *et. al.* [129] presented a statistical approach to perform trajectory computations for ice fragments that are shed from the wing and fuselage surfaces of a business jet. They carried out an experimental study of aerodynamic loads around a potential ice fragment

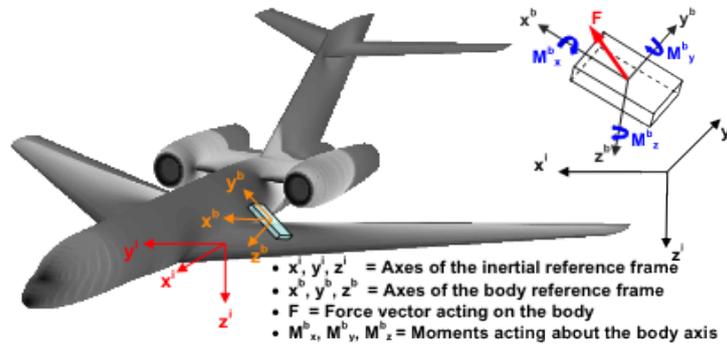


Figure 1.1 – Forces and moment on ice particle - Papadakis *et. al.* [129]

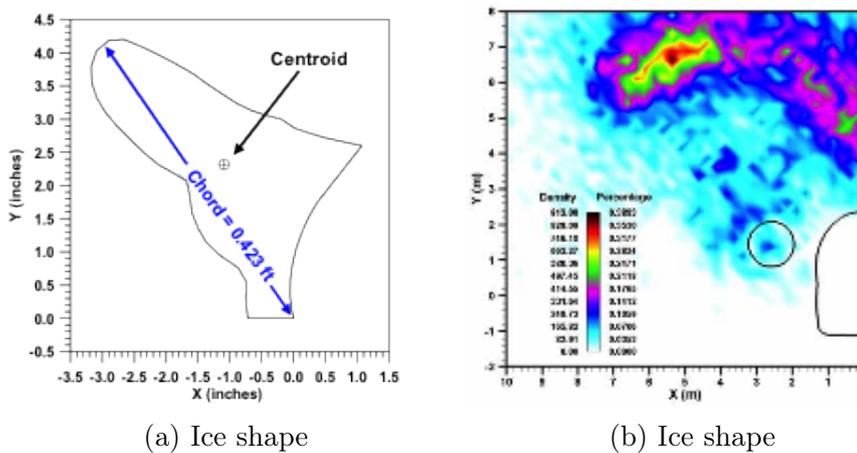


Figure 1.2 – Papadakis *et. al.* results [129]

and derived empirical correlations. A probabilistic approach has been used to identify areas where ice fragments are most likely to strike the aircraft (Monte Carlo study to take into account the uncertainties of the initial conditions and the chaotic nature of the ice particle motion). Figure 1.2a presents one of the ice shape used in [129] and figure 1.2b the results of the Monte Carlo study. A similar approach has been proposed by Deschenes *et al.* [59] where the database for the aerodynamic loads is provided by numerical simulations.

There is only very little experimental data to validate the trajectory simulations using low-fidelity models. During the master thesis of Shimoi [148] at the Wichita State University a series of different geometries were released into the free-stream of a wind tunnel test section and the comparison between simulation and experiment is convincing. There is currently no experimental data available for ice debris particles in the vicinity of an aircraft.

## 1.2.2 High fidelity models

The second type of approach, is based on a time accurate, full fluid structure interaction between the ice particle and the surrounding flow. In this state of the art, it is proposed to classify the different possibilities provided in the literature into three main kinds. The first one, that has not yet been applied to ice shedding trajectory, proposes an accurate resolution of the interaction by discretizing the solid inside the whole domain and by displacing it using some moving mesh techniques or remeshing. The second one, commonly called *overlapping grids* or *Chimera* method, is a nice alternative without remeshing. Two meshes are considered, a fine one containing an explicit discretization of the solid to solve accurately FSI, and a second one, coarser, discretizing the whole domain. Interpolations are performed from one to another at each time step. The last one, in which this work is included is called *Immersed Boundary* (IB) methods. IB methods are characterized by the absence of an explicit discretization of the solid into the mesh : the solid is then known implicitly.

### 1.2.2.1 Moving Mesh/Remeshing

When dealing with moving bodies, an idea could be to remesh at each time step according to the new position of the solid. However, this kind of approach presents some disadvantages as reported in [95]. The first one is the necessity of a remeshing software at each time step which can considerably increase the computational time. For (massively) parallel computations, the distribution of the new mesh to each processor is not an easy task and may emphasize this problem. The second one is the need of an interpolation procedure at each time step from the old mesh to the new one, that can severely impact the accuracy of the solution.

To overcome those difficulties, Johnson and Tezduyar proposed, in case of small movements, to only displace the nodes in the vicinity of the solids according to an elasticity law and they have formulated a Deformable Spatial-Domain/Stabilized-Space-Time (DSD/SST) finite element formulation. However, when displacements lead to element distortion, a remeshing procedure is necessary (this is performed only a limited number of times). Two dimensional results can be found in [95] for the study of flow past one or two oscillating naca airfoils and three dimensional simulations of multi spheres falling in a tube are reported in [96]. Based on the same idea, several works have been proposed in the literature among which the study of an elastic beam in a flow by Khurram and Masud in [99] or the study of parachute systems by Kalro and Tezduyar [97]. One can also see the work of Alauzet and co authors [16, 15].

### 1.2.2.2 Chimera methods

The idea of this method is based on overlapping grids [111], one grid to perform an accurate resolution close to the solid, and another one to solve the fluid flow on the whole domain (more than two grids can be used according to the complexity and degree of accuracy required). The usual procedure is to use interpolations of overlapping boundaries to provide the necessary communication between the different grids. However, this method has several shortcomings such as the loss of conservation, extra cost due to interpolations and locally reduced accuracy due to mismatched cell sizes between the computational meshes. However, it does not require any remeshing/moving mesh procedure.

An overset unstructured grid methodology was applied by Morton *et. al.* [119] to solve steady state and DES simulations around an aircraft C-130 H and a ringslot extraction parachute (illustration figure 1.3). The method based on unstructured hybrid overlapping grids and solving the Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations together with the flight mechanics equations has been validated for the simulation of store release trajectories interacting with a vortex flow of a military transport aircraft [145]. The simulations have been compared to dedicated wind tunnel experiments showing a good agreement for the time interval of interest. An alternative numerical approach to overset grids is proposed by Baruzzi *et. al.* [25]. At each time step, the displacement and rotation of the moving domains is computed. The moving domains containing the ice pieces are displaced and amalgamated into the fixed background mesh using a hole-cutting and stitching algorithm, see figure 1.4. The aim is to eliminate interpolations between domains and insuring flux conservation across the entire domain. In the context of the STORM project, DLR<sup>1</sup> and ONERA<sup>2</sup> propose this kind of approach as high fidelity model. Their result will be compared to the experiments performed by DLR in wind tunnel and provide a basis of comparison for the proposed approach.

### 1.2.2.3 Immersed Boundary methods

The present work belongs to the third kind of approach to deal with moving bodies that are the IB methods. This kind of method is characterized by an implicit definition of the solid. When coming to the discretization of the domain, instead of having an explicit discretization of the solid such as required by the two previous proposed approaches, the domain is discretized without considering the solids. Those solids are known *via* a level set function (see figure 1.5).

This approach was introduced by Peskin in 1971 [130] for the study of

---

<sup>1</sup>National aeronautics and space research centre of the Federal Republic of Germany

<sup>2</sup>French aerospace research center

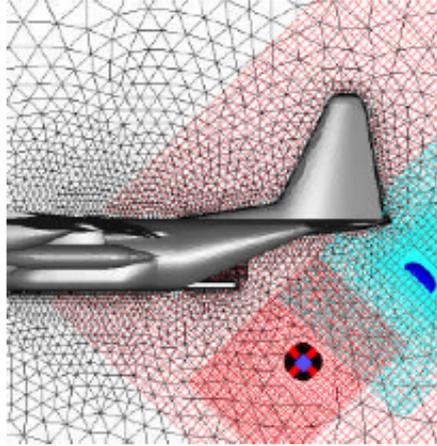


Figure 1.3 – Overset grid system for of the C-130H and extraction parachute after cargo release from Morton *et. al.* [119]

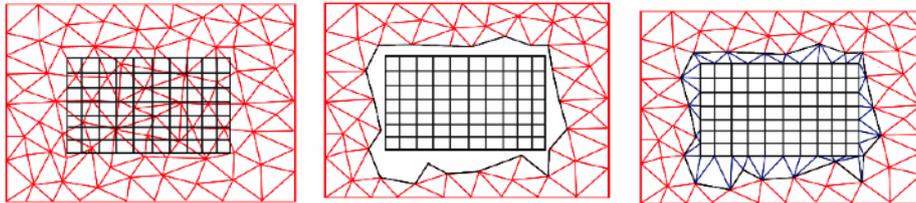


Figure 1.4 – Left : Chimera approach - Middle : Hole cutting - Right : Amalgamated domains after hole-cutting and stitching from Baruzzi *et. al.* [25]

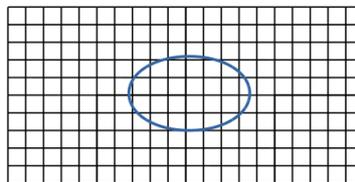


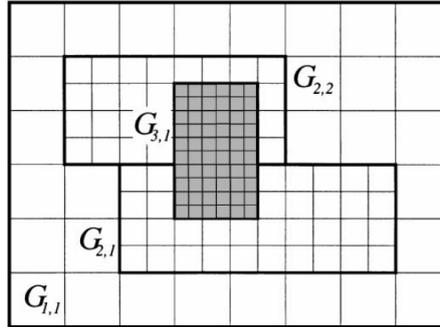
Figure 1.5 – Immersed boundary (blue) in a cartesian uniform mesh

flow around heart valves. Since then, it has become a very active field of research, mainly for the easy way of handling complex geometries and moving bodies. The initial contributions started on cartesian grids, because of the advantages provided by such grids : simplicity, ease of implementation and parallelization. Exhaustive reviews are proposed by Mittal and Iaccarino in 2005 [117] and Sotiropoulos and Yang in 2014 [150]. The large range of IB methods provided by the literature can be regrouped into two classes : *direct forcing* methods and *continuous forcing* methods.

The first category is defined by the presence of the Boundary Condition (BC) at the discrete level. The equations are discretized and only then, the BC are introduced. One of the common technique is the cut-cell finite volume approach, where the cells intersected by the IB are tracked, splitted and then merged with neighbour ones. Local geometry interface is recovered, and the schemes are conservative [44, 159, 163, 102]. However, the main drawback is the complexity involved by three dimensional cases as many geometrical patterns can be obtained for cut cells. Specific treatment is addressed with 3D results in [102]. Another technique is the ghost-cell approach. Ghost-cells are cells inside the solid that have one neighbour in the fluid. Values of the solution are extrapolated from the fluid mesh into those ghost cells so as to impose appropriate BC at the interface [113, 73, 158, 77]. One of the advantages for those two techniques is the relative ease to increase the order of accuracy to account for the BC.

The second class of IB method is characterised by the imposition of the BC *via* a source term directly in the governing equations. In the classical IB method [130], the IB enforces the BC using a smooth distribution function, whom definition has been studied to increase the accuracy of the approach [30, 143]. As mentioned in [117, 150], this formulation is better suited for elastic boundaries than rigid ones.

Another technique in the continuous forcing approach, designed for solid bodies is the *Penalization*, subject of this work. The main idea of the method and relevant references are given in the following lines, the equations will be given in section 2.1.2. The principle was initially introduced by Brinkman in 1947 [34] for a swarm of particle. In this work, the swarm of particles is assimilated to a porous mass and then ruled by the Darcy's law. Brinkman modified the incompressible Stokes equations by inserting a term such that Darcy's law is recovered for this swarm of particles. This work led to the so called *Brinkman Navier Stokes* equations or *Penalization* where a Darcy drag term is added into the equations. The solid is thus considered as a porous media with a very small permeability  $\lambda$ , the added forcing term taking the form :  $\mathbf{F} = \frac{1}{\eta}\chi_s\mathbf{u}$ , where  $\chi_s$  is the mask function of the solid. The theoretical framework has been given by Arquis and Caltagirone in [21] and more recently Angot *et. al.* performed an estimate of the error induced by penalization along with the efficiency of the method in [18]. Flow around motionless

Figure 1.6 – Structured Mesh Refinement, Roma *et. al.* [141]

bodies have been studied with such a technique on cartesian grids by Liu and Vasilyev in [110], Boiron *et. al.* in [32], and using adaptive mesh refinement (AMR) by Khadra *et. al.* [98]. Moving bodies simulations have been proposed by Kolomenskiy and Schneider [104] for falling solids, and in the context of ice shedding trajectories, this technique has been employed by Beaugendre, Morency and co author in [26, 118]. Simulations on adapted unstructured grids have been proposed by Abgrall *et. al.* in [4].

A general issue considering IB method on cartesian grids is the difficulty to define properly the IB on the background mesh. Even if high order accuracy imposition is proposed, the geometry of the interface may not be accurate. Increasing the number of nodes around the interface allows a better accuracy, however, such refinement is not straight forward on cartesian meshes. A uniform mesh refinement being inefficient, as finer areas far from the solid may not be necessary, work has been done on local Cartesian grid refinement. However, those techniques implies non conformal mesh, and thus have to be handled carefully, as proposed by Roma *et. al.* in [141] (see figure 1.6) or De Tullio *et. al.* in [57].

To overcome this difficulty IB methods have been applied recently on unstructured meshes. Indeed, this idea has become more and more popular because of the simplicity to refine area of the computational domain or to use mesh adaptation techniques. The work of Hachem *et. al.* proposed an immersed stress method on adapted meshes [82]. In [68], Farhat *et. al.* define a refined area around the considered body at the initial position. Using translation and rotation, this refinement is moved with the solid, keeping the same accuracy all along the simulation. Other approaches to deal with moving bodies involving remeshing have been proposed, such as the work of Zhou *et. al.* in [164], where triangles are divided/merged according to the position of the solid and the solution, or Jannoun's PhD [94] that proposes a remeshing technique based on metrics.

## 1.3 Description of the proposed approach

The goal of this study is to provide an efficient and accurate way of performing simulations involving moving bodies, that would provide an innovative background for high fidelity methods. The previous state of the art intends to motivate the use of an IB method, and more specifically the penalization on unstructured grids. As mentioned previously, the combination of immersed boundary method and unstructured grids gives the possibility to use mesh adaptation techniques that allows an accurate definition of the interface, with a limited number of nodes. The question that has not been addressed in the previous section 1.2 is the mesh adaptation strategy that should be adopted for such applications. Two main strategies not detailed here can be employed (non exhaustive state of the art are given in the concerned parts of this manuscript, sections 2.2 and 4.2). The first one consists in performing a remeshing of the domain (or part of the domain) and interpolation procedures update the solution from one mesh to another. This strategy has been employed in this work for solving steady simulations as powerful techniques have been already proposed in the literature (see section 2.2), and because the remeshing is performed only to increase the accuracy of the solution at the steady state, and is thus performed a limited number of times. However, when dealing with moving bodies, the same drawbacks than formulated in the previous section 1.2.2.1 are raised (necessity of remeshing, that is time consuming, and interpolation procedures at each time step, that can easily damage the accuracy of the solution). Therefore, a new approach based on a constant connectivity mesh adaptation procedure (r-adaptivity) combined to the Arbitrary Lagrangian Eulerian (ALE) framework without remeshing and interpolation steps is proposed. In addition, when coming to parallel computations, keeping the number of degrees of freedom constant allows to keep the same load balancing on the processors.

In this manuscript, to prove the potential provided by the proposed approach, laminar academic test cases are simulated before an application to a two dimensional ice shape. The final simulations propose a full fluid structure interaction, where the motions of the solids are ruled by the forces exerted on it. To simulate realistic ice shedding trajectories, the missing tool is a turbulence model to account for realistic Reynolds number.

## 1.4 Thesis Contributions and Manuscript organization

The main contributions of this PhD along with the related publications are described here. The developments have been performed in the INRIA computational code *RealFluids*, that was at the beginning of this thesis solving the

Steady Navier Stokes equations using high order residual distribution schemes (a state of the art is provided section 2.3.1). Thus, the main contributions of this work can be described as follows :

- **Residual Distribution Framework and IB** : To begin, the residual distribution framework has been extended to the penalization, in the context of steady problems. Then, the schemes proposed by De Santis in his PhD [56] have been extended to the resolution of unsteady simulations, only the second order accuracy case being considered. A splitting approach has been formulated to remove CFL constraint associated to the penalization, allowing the possibility to employ explicit schemes. Finally, ALE schemes have been proposed to solve advection diffusion problems (second order case), and applied to the resolution of the penalization in the context of moving solids.
- **r-adaptation strategy** : A combination of existing r-adaptivity techniques has been performed, leading to an approach where the mesh is assimilated to an elastic material on which forces are applied according to the localisation of the embedded boundary and physical phenomena.

The publication and conferences reporting the results obtained during this work are listed hereafter :

- Journals :
  - L. Nouveau, H. Beaugendre, C. Dobrzynski, R. Abgrall, M. Ricchiuto. An adaptive, residual based, splitting approach for the penalized Navier Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 303 : 208-230. 2016. <http://www.sciencedirect.com/science/article/pii/S0045782516300019>
  - R. Abgrall, H. Alcin, C. Dobrzynski, H. Beaugendre, L. Nouveau. Residual Schemes Applied to an Embedded Method Expressed on Unstructured Adapted Grids. *Acta Aerodynamica Sinica*, 34(02) p214-223, 2016. <http://html.rhhz.net/KQDLXXB/2016-02-214.htm>
- Proceedings :
  - L. Nouveau, R. Abgrall, H. Alcin, H. Beaugendre, and C. Dobrzynski. Residual Distribution Schemes for Penalized Navier Stokes Equations on Adapted Grids, *ECCM V - ECFD VI*. Barcelona, Spain, July 2014.
  - R. Abgrall, H. Alcin, H. Beaugendre, C. Dobrzynski, and L. Nouveau. An adaptive ALE residual based penalization approach for laminar flows with moving bodies. *8<sup>th</sup> International Conference on Computational Fluid Dynamics*. ICCFD8-2014-0166, Chengdu, China, July 2014

- Technical Report :
  - L. Nouveau, H. Beaugendre, M. Ricchiuto, C. Dobrzynski, R. Abgrall. An adaptive ALE residual based penalization approach for laminar flows with moving bodies. RR N 8936. INRIA, 2016. <https://hal.inria.fr/hal-01348902>
- Conferences :
  - L. Nouveau, M. Ricchiuto, H. Beaugendre, C. Dobrzynski, R. Abgrall. An adaptive, residual based splitting approach for the time dependent penalized Navier Stokes equations. ECCOMAS VII, 2016, Hersonissos, Creete.
  - L. Nouveau, M. Ricchiuto, H. Beaugendre, C. Dobrzynski, R. Abgrall. An ALE residual distribution approach applied to the penalized Navier Stokes equations on adapted grids for moving solids. CANUM 2016, Obernai, France.
  - L. Nouveau, R. Abgrall, H. Alcin, H. Beaugendre, C. Dobrzynski. Residual Distribution Schemes for Penalized Navier Stokes Equations on Adapted Grids. ECCOMAS VI, 2014, Barcelona, Spain.

The Manuscript is organized into three chapters. The first one provides results concerning steady penalized results on adapted mesh and is divided into three main topics. To begin, section 2.1 gives a formulation of the penalized Navier Stokes equations. Then, the mesh adaptation strategy adopted for steady simulation, that is a metric mesh based adaptation, is presented in section 2.2. Finally, in the last section 2.3, the resolution of the equations using a residual based approach on adapted grids is proposed .

The second chapter is devoted to the resolution of unsteady simulations, but considering motionless bodies. In a first section 3.1, the schemes employed are extended to unsteady simulations, starting from a scalar conservation law. Convergence studies are performed for a scalar problem and Navier Stokes equations. Then, in section 3.2, the splitting approach used to solve unsteady penalized Navier Stokes equations is presented. Simulations on fixed grids adapted only at the interface fluid/structure are proposed to validate the method.

In the last chapter, the approach for simulations involving moving bodies is presented. In section 4.1, Arbitrary Lagrangian Eulerian methodology is proposed, beginning with a recall on the principle. Then a formulation of the schemes in this framework, from the scalar advection to the Navier Stokes equations is proposed. The r-adaptation strategy employed is described in section 4.2. Finally, penalized simulation involving moving bodies are presented in section 4.3 to demonstrate the advantages proposed by the present study.

A final chapter will conclude and present the opportunities opened by this work.



## Chapter 2

# Steady Penalized Navier Stokes Equations

The focus of this first chapter is the methodology employed to solve the penalized Navier Stokes equations for steady problem. After defining the equations involved, the metric based mesh adaptation strategy is presented. A (non exhaustive) state of the art regrouping the main contribution to this field is proposed and a recall of the theory for our specific application is given. The numerical schemes, that are the residual distribution schemes, used to discretize the equations are then defined. A state of the art referencing the different contributions and more specifically the ones concerning the context of advection diffusion problems is provided. Those schemes are then extended to the penalization theory and some results proving the ability of the combination penalization/mesh adaptation to provide competitive results in comparison with "classical" approach are given.

## 2.1 Problem Statement

In this section, the equations ruling the considered problem are presented. The Navier Stokes (NS) system of equations is described, and the way of introducing Boundary Conditions (BC) for classical fitted resolutions is recalled, to introduce the penalized NS equations, subject of this PhD. A presentation of the dimensionless formulation of the problem is then provided. Finally, scalar conservation laws are presented, simplified model of the problem, that will be used to develop numerical tools.

### 2.1.1 Physical Model, Navier Stokes equations

#### 2.1.1.1 Navier Stokes Equations

The dynamic of a compressible flow taking into account the viscous and thermal diffusion is described by the system of *Navier Stokes* (NS) equations, composed with the following set of equations :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \\ \frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + p \mathbb{I}) = \nabla \cdot \mathbb{S} \\ \frac{\partial(\rho e)}{\partial t} + \nabla \cdot ((\rho e + p) \mathbf{v}) = \nabla \cdot (\mathbb{S} \mathbf{v} - \mathbf{q}) \end{cases} \quad (2.1)$$

$\rho$  denotes the density,  $\mathbf{v}$  the vector of velocities,  $p$  the pressure and  $e$  the total energy.  $\mathbb{I}$  denotes the identity matrix,  $\mathbb{S}$  is the viscous stress tensor and  $\mathbf{q}$  is the heat flux.

For a Newtonian fluid, the stress tensor is given by :

$$\mathbb{S} = \lambda \text{tr}(\mathbb{E}) + 2\mu \mathbb{E} \quad (2.2)$$

where  $\mu$  is the viscosity and  $\lambda$  linked to  $\mu$  by  $3\lambda + 2\mu = 0$ .  $\mathbb{E}$  is the tensor of deformation :

$$\mathbb{E} = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \quad (2.3)$$

The heat flux is ruled by the Fourier law :

$$\mathbf{q} = -\kappa \nabla T \quad (2.4)$$

where  $T$  is the temperature and  $\kappa$  the thermal conductivity. In the system (2.1), the first equations corresponds with the *mass conservation* equation, the second with the *momentum conservation* equations and the third one with the *energy conservation* equation.

Those equations need additional relations to close the system, that will be defined by the choice done to describe the gas. The simplest way is to consider perfect gas, that is a good approximation of real gas behaviour at low pressure and high temperature. Some more complex state laws could be use, out of the scope of this PhD (see for instance [5] where Peng–Robinson–Stryjek–Vera and Span–Wagner models are employed to perform simulations).

The internal energy  $\epsilon$  is linked to the temperature  $T$  via the constant gas  $\mathcal{R}$  and the specific heat ratio  $\gamma = c_p/c_v$ ,  $c_p$  and  $c_v$  being the specific heats at constant pressure and volume :

$$\epsilon = c_v T = \frac{\mathcal{R}}{\gamma - 1} T \quad (2.5)$$

The pressure  $p$  is linked to the density  $\rho$  via :

$$p = \rho \mathcal{R} T \quad (2.6)$$

and the sound velocity  $c$  is :

$$c = \sqrt{\frac{\gamma P}{\rho}} \quad (2.7)$$

The internal energy is linked to the total energy and the velocity by :

$$e = \epsilon + \frac{\|v\|^2}{2} \quad (2.8)$$

This system of equations (2.1) is commonly expressed over a matrix conservative form :

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}^{Eul}(\mathbf{u}) = \nabla \cdot \mathbf{G}^{NS}(\mathbf{u}, \nabla \mathbf{u}) \quad (2.9)$$

where  $\mathbf{u}$  is the vector of conservative variables,  $\mathbf{F}^{Eul}$  is the Euler (advective) flux and  $\mathbf{G}^{NS}(\mathbf{u}, \nabla \mathbf{u})$  is the viscous flux, which depends not only on  $\mathbf{u}$ , but also on its gradient  $\nabla \mathbf{u}$ . The Euler Jacobian  $\mathbb{A} = \nabla_{\mathbf{u}} \mathbf{F}^{Eul}$  is introduced and the viscous flux can be written over the form :

$$\mathbf{G}^{NS}(\mathbf{u}, \nabla \mathbf{u}) = \mathbb{K}(\mathbf{u}) \nabla \mathbf{u} \quad (2.10)$$

leading to the following expression of the NS system of equations :

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbb{A} \cdot \nabla \mathbf{u} - \nabla \cdot (\mathbb{K}(\mathbf{u}) \nabla \mathbf{u}) = 0 \quad (2.11)$$

$\mathbb{A}$  and  $\mathbb{K}$  are tensors. We denote  $m$  the number of equations and  $d$  the dimension ( $m = 2 + d$ ), and as laminar flows are considered, no additional equations associated to a turbulent model are introduced. Thus, the sizes of  $\mathbb{A}$  and  $\mathbb{K}$  are respectively  $m \times m \times d$  and  $m \times d \times m \times d$ . Detailed expressions of those tensors are given appendix A.

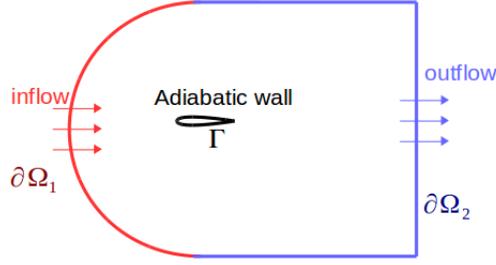


Figure 2.1 – Boundary Condition

### 2.1.1.2 Boundary Conditions

When considering the resolution of NS problems, additional equations are introduced to take into account the presence of obstacle, or to simulate infinite domain. Inflow and outflow are usually employed to model the physics on the boundary of the computational domain, and the modelisation of the solid needs to be characterized according to the willing simulations. For instance, looking at figure 2.1, three boundary conditions need to be imposed. The surface  $\Gamma$  delimiting the obstacle must provide informations w.r.t. the characteristic of the solid. In the present example, adiabatic boundary conditions are required, characterized by null velocity and normal heat flux, defined by the following set of equations :

$$\begin{cases} \mathbf{v}|_{\Gamma} = \mathbf{0} \\ \mathbf{q} \cdot \hat{\mathbf{n}}|_{\Gamma} = 0 \end{cases} \quad (2.12)$$

where  $\hat{\mathbf{n}}$  corresponds to the normalized normal.

As said in the introduction, the penalization is an Immersed Boundary (IB) method. Such additional equations are not required (at least for the solid defined by the IB), the boundary conditions (BC) being directly modelled inside the set of equations, as presented in the next section.

## 2.1.2 Penalized Navier Stokes equations

### 2.1.2.1 Modification of the equations

This work deals with the IB method called Penalization. Thus, when coming to the discretization of the problem, the solids are not explicitly present on the mesh and the BCs are imposed in a different way. For penalization, the principle is to consider the BCs by adding a volumetric source term directly into the equations. It modifies the Navier Stokes into the so-called *Penalized*

or *Brinkman* Navier Stokes equations [18, 4] :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \\ \frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + p \mathbb{I}) + \frac{1}{\eta} \sum_{i=1}^{N_s} \chi_{S_i} \rho (\mathbf{v} - \mathbf{v}_{S_i}) = \nabla \cdot \mathbb{S} \\ \frac{\partial(\rho e)}{\partial t} + \nabla \cdot ((\rho e + p) \mathbf{v}) + \frac{1}{\eta} \sum_{i=1}^{N_s} \chi_{S_i} \rho [(\mathbf{v} - \mathbf{v}_{S_i}) \cdot \mathbf{v} + \theta_{S_i} (\epsilon - \epsilon_{S_i})] = \nabla \cdot (\mathbb{S} \mathbf{v} - \mathbf{q}) \end{cases} \quad (2.13)$$

$N_s$  is the number of solids and  $\chi_{S_i}$  their characteristic functions.  $\mathbf{v}_{S_i}$ ,  $\epsilon_{S_i}$  and  $T_{S_i}$  correspond to the velocities, energies and temperatures of the solids.  $\theta_{S_i}$  is set to 0 or 1 so as to impose or not condition on the temperature.  $\eta$  is the *penalty parameter* and chosen very small ( $\frac{1}{\eta} \gg 1$ ). The accuracy of the method depends on the value of  $\eta$  [18]. In the simulations, the parameter is set to  $\eta = 10^{-10}$ . The matrix form of the system (2.13) writes :

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}^{Eul}(\mathbf{u}) + \mathbf{S} &= \nabla \cdot \mathbf{G}^{NS}(\mathbf{u}, \nabla \mathbf{u}) \\ \mathbf{S} &= \frac{1}{\eta} \sum_{i=1}^N \rho \chi_{S_i} \begin{pmatrix} 0 \\ \mathbf{v} - \mathbf{v}_{S_i} \\ (\mathbf{v} - \mathbf{v}_{S_i}) \cdot \mathbf{v} + \theta_{S_i} (\epsilon - \epsilon_{S_i}) \end{pmatrix} \end{aligned} \quad (2.14)$$

The question is now the definition of the characteristic function  $\chi_{S_i}$ . In this work, the *Signed Distance Function* is employed to capture on the mesh the position of the solid.

### 2.1.2.2 Signed Distance Function

The signed distance function corresponds to the distance of the considered point of the mesh to the explicit surface of the object with a sign to precise if it is an inner or outer point. It can be define as follows :

#### Definition 2.1. Signed Distance Function

Let  $\Omega_1 \subset \Omega_2$  delimited by the surface  $\Gamma$ . The signed distance function  $\psi(\mathbf{x})$  is defined as :

$$\psi(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_2 \setminus \Omega_1 \\ -d(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_1 \end{cases} \quad (2.15)$$

with  $d$  the distance from the point  $\mathbf{x}$  to the surface  $\Gamma$  (see figure 2.2).

This signed distance function can be computed in several way. Here we require the SDF to be negative inside the solid and positive outside. Thus, let define the inside of the solid by a domain  $\Omega$  :

$$\Omega = \{\mathbf{x} \in \mathbb{R}^d | \psi_0(\mathbf{x}) < 0\} \text{ and } \partial\Omega = \{\mathbf{x} \in \mathbb{R}^d | \psi_0(\mathbf{x}) = 0\} \quad (2.16)$$

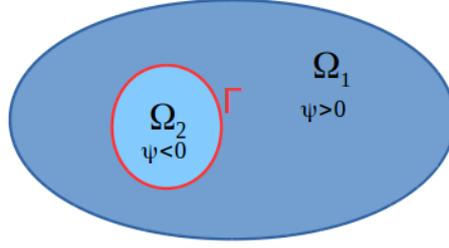


Figure 2.2 – Signed Distance Function

with  $d$  the number of space dimensions and  $\psi_0$  a continuous function. The SDF is the solution of the *unsteady Eikonal equation* :

$$\begin{cases} \frac{\partial \psi}{\partial t} + \text{sgn}(\psi_0)(\|\nabla \psi\| - 1) = 0, \forall t > 0, \mathbf{x} \in \mathbb{R}^d \\ \psi(t = 0, \mathbf{x}) = \psi_0(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^d \end{cases} \quad (2.17)$$

This can be solved with several methods. The one chosen here is the one proposed by Dapogny and Frey in [55] based upon the characteristic method giving the approximated solution :

$$\psi \approx \begin{cases} \psi_0 \left( \mathbf{x} - dt \frac{\nabla \psi_0}{\|\nabla \psi_0\|} \right) + dt, & \text{for } \mathbf{x} \in \overline{c\Omega} \\ \psi_0 \left( \mathbf{x} + dt \frac{\nabla \psi_0}{\|\nabla \psi_0\|} \right) - dt, & \text{for } \mathbf{x} \in \Omega \end{cases} \quad (2.18)$$

### 2.1.3 Dimensionless form of the equations

For numerical purposes, the equations are written in a dimensionless form. The process is the following : for each physical variable  $V_p$  we set :

$$V_p = \tilde{V}_p V_{p_r} \quad (2.19)$$

where  $\tilde{V}_p$  is the dimensionless variable and  $V_{p_r}$  is the reference variable. Let  $L_r, v_r, t_r$  be the reference length, velocity and time linked by  $v_r = L_r/t_r$ . We can define the following dimensionless operators :

$$\frac{\partial}{\partial \tilde{t}} = \frac{L_r}{v_r} \frac{\partial}{\partial t} \quad \tilde{\nabla} = L_r \nabla \quad (2.20)$$

and variables :

$$\begin{aligned} \tilde{\rho} &= \frac{\rho}{\rho_r}, & \tilde{\mathbf{u}} &= \frac{\mathbf{u}}{v_r}, & \tilde{e} &= \frac{e}{e_r}, & \tilde{T} &= \frac{T}{T_r}, & \tilde{p} &= \frac{p}{p_r} \\ \tilde{\mu} &= \frac{\mu}{\mu_r}, & \tilde{\kappa} &= \frac{\kappa}{\kappa_r}, & \tilde{\mathcal{R}} &= \frac{\mathcal{R}}{\mathcal{R}_r}, & \tilde{c}_v &= \frac{c_v}{c_{v_r}}, & \tilde{c}_p &= \frac{c_p}{c_{p_r}} \end{aligned} \quad (2.21)$$

## 2. Steady Penalized Navier Stokes Equations

---

With those definitions, we can recast the penalized NS equations as :

$$\left\{ \begin{array}{l} \frac{\partial \tilde{\rho}}{\partial t} + \tilde{\nabla} \cdot (\tilde{\rho} \tilde{\mathbf{u}}) = 0 \\ \frac{\partial (\tilde{\rho} \tilde{\mathbf{u}})}{\partial t} + \tilde{\nabla} \cdot \left( \tilde{\rho} \tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}} + \left[ \frac{p_r}{\rho_r v_r^2} \right] \tilde{p} \mathbf{I} \right) + t_r \frac{1}{\eta} \sum_{i=1}^{N_S} \chi_{S^i} (\tilde{\rho} \tilde{\mathbf{u}} - \tilde{\rho} \tilde{\mathbf{u}}_{S^i}) = \frac{1}{Re_r} \tilde{\nabla} \cdot \tilde{\boldsymbol{\sigma}} \\ \frac{\partial (\tilde{\rho} \tilde{e})}{\partial t} + \tilde{\nabla} \cdot \left( (\tilde{\rho} \tilde{e} + \left[ \frac{p_r}{\rho_r e_r} \right] \tilde{p}) \tilde{\mathbf{u}} \right) + t_r \frac{1}{\eta} \sum_{i=1}^{N_S} \chi_{S^i} [\theta_{S^i} (\tilde{\rho} \tilde{e} - \tilde{e}_{S^i}) + (\tilde{\rho} \tilde{\mathbf{u}} - \tilde{\rho} \tilde{\mathbf{u}}_{S^i}) \cdot \tilde{\mathbf{u}}] = \\ \frac{1}{Re_r} \tilde{\nabla} \cdot \left( \left[ \frac{v_r^2}{e_r} \right] \tilde{\boldsymbol{\sigma}} \tilde{\mathbf{u}} - \frac{1}{Pr_r} \left[ \frac{c_{p_r} T_r}{e_r} \right] \tilde{\mathbf{q}} \right) \end{array} \right. \quad (2.22)$$

where

$$Re_r = \frac{\rho_r v_r L_r}{\mu_r}, \quad Pr_r = \frac{\mu_r c_{p_r}}{\kappa_r} \quad (2.23)$$

are the reference Reynolds and Prandtl numbers. To reduce the number of references quantities to choose, we require both relations between square brackets and the reference Prandtl number to satisfy :

$$\frac{p_r}{\rho_r v_r^2} = 1, \quad \frac{v_r^2}{e_r} = 1, \quad Pr_r = 1 \quad (2.24)$$

Obviously, the closure equations that are here the Perfect Gas law also need to be rewritten in dimensionless form. The aim is to limit the number of reference variables to 4. Here the choice made is to use the following set of reference quantities :

$$(\rho_r, v_r, T_r, L_r)$$

For constant viscosity, we also set :  $\mu_r = \mu$ , leading to  $\tilde{\mu} = 1$ . Now, let us rewrite the different relations coming from the PG law. First, we consider relation (2.6) :

$$p = \rho \mathcal{R} T \Rightarrow \tilde{p} p_r = \rho \tilde{\rho} \tilde{\mathcal{R}} \mathcal{R}_r \tilde{T} T_r \\ \tilde{p} = \left[ \frac{\rho_r \mathcal{R}_r T_r}{p_r} \right] \tilde{\rho} \tilde{\mathcal{R}} \tilde{T}$$

By imposing that the fraction between bracket is equal to one, we obtain :

$$\mathcal{R}_r = \frac{p_r}{\rho_r T_r} = \frac{v_r^2}{T_r} \quad (2.25)$$

The reference energy  $e_r (= v_r^2$  from (2.24)) is used to compute the dimensionless internal energy  $\tilde{e}$ . Thus equation (2.5) writes :

$$\epsilon = c_v T \Rightarrow \tilde{e} v_r^2 = \tilde{c}_v c_{v_r} \tilde{T} T_r \\ \Rightarrow \tilde{e} = \left[ \frac{c_{v_r} T_r}{v_r^2} \right] \tilde{c}_v \tilde{T}$$

Requiring the bracketed quantity to be equal to one, we obtain  $c_{v,r} = \mathcal{R}_r$  (using 2.25), which gives the following dimensionless specific heats :

$$\tilde{c}_v = \frac{\tilde{\mathcal{R}}}{\gamma - 1}, \quad \tilde{c}_p = \frac{\gamma \tilde{\mathcal{R}}}{\gamma - 1} \quad (2.26)$$

The speed sound calculated from pressure and density (2.7) can be written as :

$$\begin{aligned} \tilde{c}v_r &= \sqrt{\frac{\gamma \tilde{p} p_r}{\tilde{\rho} \rho_r}}, \text{ and so} \\ \tilde{c} &= \frac{1}{v_r} \sqrt{\frac{p_r}{\rho_r}} \sqrt{\frac{\gamma \tilde{p}}{\tilde{\rho}}} \end{aligned}$$

leading to (using (2.24)) :

$$\tilde{c} = \sqrt{\frac{\gamma \tilde{p}}{\tilde{\rho}}} \quad (2.27)$$

Finally, the dimensionless thermal conductivity is defined. Recalling that the reference Prandtl number is set to 1 (2.24) :  $\kappa_r = \mu c_{p,r}$  and thus :

$$\begin{aligned} \tilde{\kappa} &= \frac{\kappa}{\kappa_r} = \frac{\kappa}{\mu c_{p,r}} = \frac{\kappa}{\mu c_p} \tilde{c}_p \\ \tilde{\kappa} &= \frac{\tilde{c}_p}{Pr} \end{aligned} \quad (2.28)$$

with  $Pr$  the Prandtl number  $Pr = \frac{\mu c_p}{\kappa}$ .

The final set of relations in dimensionless form is :

$$\begin{aligned} \tilde{p} &= \tilde{\rho} \tilde{\mathcal{R}} \tilde{T}, \quad \tilde{\epsilon} = \tilde{c}_v \tilde{T}, \quad \tilde{\kappa} = \frac{\tilde{c}_p}{Pr} \\ \tilde{c} &= \sqrt{\frac{\gamma \tilde{p}}{\tilde{\rho}}}, \quad \tilde{c}_p = \gamma \tilde{c}_v = \frac{\gamma \tilde{\mathcal{R}}}{\gamma - 1} \end{aligned} \quad (2.29)$$

### 2.1.4 Simplified Model - Scalar Conservation Law

The Navier Stokes system of equations is quite complex to solve. A common procedure when developing numerical tools is to begin with simplified model problems. For instance, scalar conservation laws can be employed as a general framework. During this work, new developments, such as the extension of the schemes to unsteady problems or in the ALE framework, have been initially performed for a scalar advection equation of the form :

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathcal{F}(u) = 0 \quad (2.30)$$

where  $\mathcal{F}$  is the advective flux. Assuming a differentiable solution, the Jacobian of this flux can be introduced  $\mathbf{a}(u) = \nabla_u \mathcal{F}$  and the equation can be written :

$$\frac{\partial u}{\partial t} + \mathbf{a}(u) \cdot \nabla u = 0 \quad (2.31)$$

Two cases can be considered here, the first one is the linear advection, and  $\mathbf{a}$  is a constant vector. The second possibility is to define a non linear flux and the advection velocity is in this case dependent on the conservative variable  $\mathbf{u}$ . A well known example is the Burger equation,  $\mathbf{a}(u) = (u, u)^T$ .

The advection diffusion can then be defined by :

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathcal{F}(u) = \nabla \cdot (\underline{\mathbf{K}} \nabla u) \quad (2.32)$$

where  $\underline{\mathbf{K}}$  is a diffusion matrix. This latest equation (2.32) is considered as a simplified version of the NS system. Concerning the diffusive part, the simplest case is to consider an isotropic diffusion, defining the matrix diffusion  $\underline{\mathbf{K}}$  homogeneous to the identity matrix :  $\underline{\mathbf{K}} = \nu \mathbb{I}$ ,  $\nu$  being associated to a viscosity. Thus, a non linear advection diffusion problem writes :

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathcal{F}(u) = \nu \Delta u \quad (2.33)$$

## 2.2 Mesh Adaptation for Steady Simulation

### 2.2.1 State of the art

For steady simulations, a metric based adaptation is employed. As explained in the introduction, the use of IB technique on unstructured grids is mainly motivated by the possibility to perform mesh refinement or mesh adaptation. Indeed, the accuracy of the imposition of the BC is directly linked to the degree of precision the solid is implicitly described. Thus, the first aim of the mesh adaptation is to propose a refined area close to the interface so as to have a precise definition of the solid *via* its characteristic function. In addition, we adapt with respect to (w.r.t.) variations of a chosen physical variable to improve the resolution of the physics and accurately capture some physical phenomena (*e.g.* shocks) with a limited CPU time overhead. This mesh adaptation technique aims at controlling an error (of the approximated solution for instance when dealing with physical adaptation) by modifying the mesh. The basic principle is to find a good error estimator to be able to prescribe sizes and directions at the nodes of the mesh. It is very well suited to generated *anisotropic* meshes, characterized by their possibility to have stretched elements if different sizes are imposed in the different directions. Anisotropy is very efficient because allows to reduce a lot the number of nodes in the refined area with respect to isotropy. Especially when considering a shock or an interface (as for the present study, a 0 level set refinement), as illustrated figure 2.3.

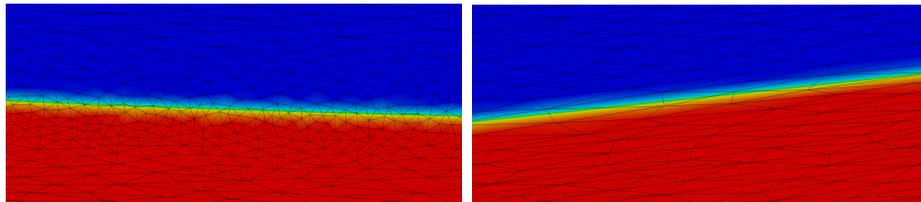


Figure 2.3 – 0 level set adaptation - Right : isotropic, Left : anisotropic

For physical adaptation, Frey, Alauzet and co authors propose an adaptation based on the Hessian of the solution. One can refer to [70] for CFD computations or [62] for buildings aero heating application. For interface adaptation, Frey *et. al.* proposed a method based on the curvature, see for instance [65, 55], and [37, 81] for application to multifluid flows problems. We refer to [17] for a recent review of the current status of mesh adaptation and its applications.

The same Hessian based error estimate has been employed at the Imperial College of London for geofluid problems [76, 131], Boltzmann transport equation resolution [22] and unsteady CFD problems [127].

Another approach has been proposed by Coupez, Hachem and co authors, that is an edge based interpolation estimate error, where the aim is to avoid the computation of the Hessian and to define an error estimator using the gradients and edge length. See [49] for application to interface adaptation, and [50] for incompressible simulations. Level set specific adaptation has been proposed with application to IB method where the length are defined *a priori* according to the distance of the solid in [82], which has been extended recently to the generation of boundary layer in [31].

In this work, the Hessian approach is employed for physical adaptation and the adaptation of the 0 level set is based on the curvature of the surface. In the results/illustrations that will be presented, the MMG library [63, 54] is employed. In this chapter, we first recall some notions about metrics and how to use them for mesh adaptation before defining the ones used for physical and level set adaptations. Then, the process adopted for the simulations leading to a convergence mesh solution is presented.

### 2.2.2 Some metric Notions

We begin with the definition of a *metric tensor* (or metric) :

**Definition 2.2. Metric**

A metric  $\mathcal{M}$  in  $\mathbb{R}^n$  is a  $n \times n$  symmetric definite positive (SDP) matrix.

Being SDP, this matrix is diagonalizable :  $\mathcal{M} = \mathcal{R}^T \Lambda \mathcal{R}$  where  $\mathcal{R}$  is the eigenvectors matrix and  $\lambda$  the eigenvalues matrix. The eigenvalues are denoted  $(\lambda_i)_{i=1,n}$ . Considering two vectors  $\mathbf{u}$  and  $\mathbf{v}$  in  $\mathbb{R}^d$ , the scalar product associated to the metric is defined as :

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} := \mathbf{u}^T \mathcal{M} \mathbf{v} \quad (2.34)$$

The norm of the vector  $\mathbf{u}$  can then be defined :

$$\|\mathbf{u}\|_{\mathcal{M}} := \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} = \sqrt{\mathbf{u}^T \mathcal{M} \mathbf{u}}$$

$\|\mathbf{u}\|$  measures the length of the vector  $\mathbf{u}$  with respect to the metric  $\mathcal{M}$ . A metric  $\mathcal{M}$  can be represented by an ellipsoid  $\mathcal{E}_{\mathcal{M}} = \{\mathbf{P}, \|\mathbf{OP}\|_{\mathcal{M}} = 1\}$ ,  $\mathbf{O}$  being the center of this ellipsoid. The directions of the ellipsoid are given by the eigenvectors and the sizes are linked to the eigenvalues with  $h_i = \frac{1}{\sqrt{\lambda_i}}$  (see figure 2.4). Then, the definition of *Euclidean metric space* can be provided :

**Definition 2.3. Euclidean metric space**

An Euclidean metric space  $(\mathbb{R}^d, \mathcal{M})$  is a vector space supplied with a scalar product associated to a metric  $\mathcal{M}$  defined by (2.34).

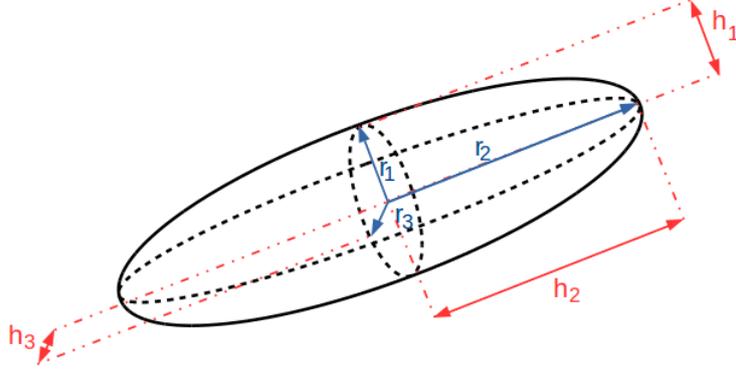


Figure 2.4 – Ellipsoid representation of a metric

In such a space, the length between two points  $\mathbf{M}$  and  $\mathbf{N}$   $l_{\mathcal{M}}(\mathbf{MN})$  is the distance between those two points :

$$l_{\mathcal{M}}(\mathbf{MN}) = \sqrt{\langle \mathbf{MN}, \mathbf{MN} \rangle_{\mathcal{M}}} = \sqrt{(\mathbf{MN})^T \mathcal{M} \mathbf{MN}} \quad (2.35)$$

For the definition of Euclidean metric space, the metric is constant over the space. The more general case where the metric, and so the scalar product, vary all over the domain lead to the definition of *Riemannian metric space* :

**Definition 2.4. Riemannian metric space**

A *Riemannian metric space* is a manifold  $\Omega \subset \mathbb{R}^d$  supplied with a smooth metric  $\mathcal{M}(\cdot)$ .

The length between two points  $\mathbf{M}$  and  $\mathbf{N}$  is this time no more the length of the vector  $\mathbf{MN}$  associated to a unique metric  $\mathcal{M}$  but defined as :

$$l_{\mathcal{M}}(\mathbf{MN}) = \int_0^1 \sqrt{(\mathbf{MN})^T \mathcal{M}(\mathbf{M} + t\mathbf{MN}) \mathbf{MN}} dt \quad (2.36)$$

## 2.2.3 Metric Expression For Mesh Adaptation

### 2.2.3.1 Principle

To generate an anisotropic mesh, we specify at each point of the initial grid the sizes and directions of the edges. For this purpose, at each node, a metric is defined containing those informations. The sizes  $h_i$  are linked to the eigenvalues with  $\lambda_i = \frac{1}{h_i^2}$  and the eigenvectors corresponds with the directions (the ellipsoid is exactly defined by the size and direction wanted for the mesh). Thus, the idea is to work in a Riemannian metric space whose associated metric field is the one containing mesh informations. What is called a *unit mesh* in this specific space is then sought, *ie* the length of the edges  $\mathbf{e}$  computed in the

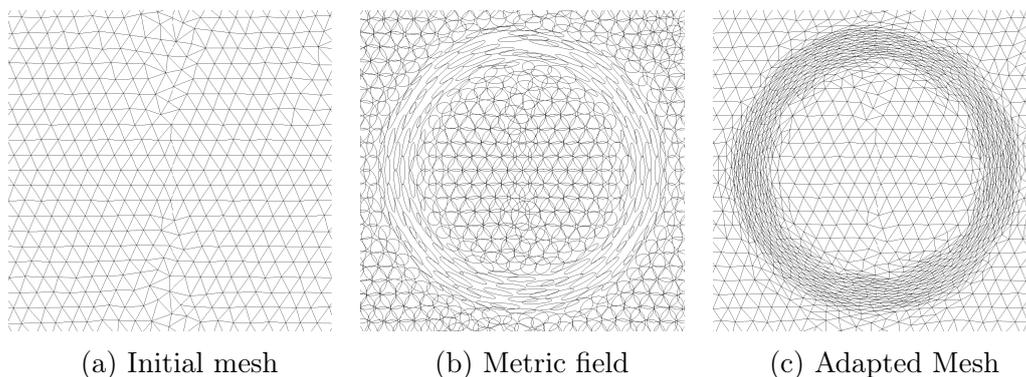


Figure 2.5 – Mesh Adaptation and Metrics

space are equal to one :  $l_{\mathcal{M}}(\mathbf{e}) = 1$ . In practice, such meshes are impossible to generate. Thus, a mesh as close as possible to the unit mesh is generated. Figure 2.5 sketches the idea for the mesh adaptation : 2.5a corresponds to the initial mesh and figure 2.5b presents the computed metric field on this mesh. Figure 2.5c shows the resulted adapted mesh.

Now the question is how to construct the metric corresponding to the adaptations required *i.e.* how define properly the eigenvalues and eigenvectors of the metrics. The two next sections are dedicated to the definition of the metric for an adaptation to a physical variable (density, pressure, etc...) and for the SDF adaptation.

### 2.2.3.2 Physical Adaptation

For the physical adaptation, the *approximation error* is considered. It corresponds to the error between the approximated solution  $u_h$  resulting from the numerical simulation on the mesh  $\mathcal{T}^h$  and the exact solution of the problem  $u$  :  $e_h = \|u - u_h\|$ ,  $\|\cdot\|$  being a norm of  $\mathcal{R}^d$ ,  $d$  the dimension. The idea is to equidistribute this error by adding nodes in large variation of the solution (*e.g.* shocks) and to remove some where the solution remains uniform. We just recall here the main step leading to the definition of the metric defining an upper bound of this error, the full demonstration being proposed in [70]. The starting point is Cea's lemma that gives as upper bound of the approximation error the *interpolation error* :

$$e_h \leq C \|u - \Pi_h u\|_{H^1} \quad (2.37)$$

where  $\Pi_h u$  is the linear interpolate of  $u$  on the mesh  $\mathcal{T}^h$  and  $C$  a constant independent of the mesh. As explained in [70], Cea's Lemma is demonstrated for elliptic problem but assumed also true for the considered one (*e.g.* hyperbolic). Thus, the aim is now to bound this interpolation error for each

element  $T$  of  $\mathcal{T}^h$ . Considering the  $L^\infty$  norm in (2.37), the following theorem (a demonstration being provided in [70]), involving the Hessian of the solution  $\mathcal{H}_u$ , holds :

**Theorem 2.1.** *Upper bound of the interpolation error on element*

$$\|u - \Pi_h u\|_{\infty, T} \leq c_d \max_{\mathbf{x} \in T} \max_{\mathbf{e} \in T} \langle \mathbf{e}, |\mathcal{H}_u(\mathbf{x})| \mathbf{e} \rangle \quad (2.38)$$

where  $c_d$  is a constant depending only on the dimension.

However, this inequality is not enough because  $\max_{\mathbf{x} \in T} (\mathcal{H}_u)$  is not known and difficult to compute numerically. Nevertheless, an upper bound of the right hand side of the inequality (2.38) can be provided using metrics :

$$\max_{\mathbf{x} \in T} \max_{\mathbf{e} \in T} \langle \mathbf{e}, |\mathcal{H}_u(\mathbf{x})| \mathbf{e} \rangle \leq \langle \mathbf{e}, \mathcal{M}(T) \mathbf{e} \rangle \quad (2.39)$$

Such metrics are defined as follows :

$$\mathcal{M} = \mathcal{R} \tilde{\Lambda} \mathcal{R}^{-1}, \quad \tilde{\Lambda} = \begin{pmatrix} \tilde{\lambda}_1 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 \\ 0 & 0 & \tilde{\lambda}_3 \end{pmatrix} \quad (2.40)$$

$$\tilde{\lambda}_i = \min \left( \max \left( \frac{c_d |\lambda_i|}{\epsilon}, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right)$$

where  $\mathcal{R}$  is the matrix of the eigenvectors of the Hessian  $\mathcal{H}_u$  and  $\lambda_i$  the eigenvalues.  $\epsilon$  is the wanted error.  $h_{max}$  and  $h_{min}$  are respectively the minimum and maximum size wanted for the mesh. Indeed, without such a truncation, in area of constant solution, infinite length elements would be computed, or on the opposite, in areas of large variations of the solution, element sizes of little use from a computational point of view would be imposed.

### 2.2.3.3 Adaptation to the Signed Distance

As proposed in [65] (among others), the error to control for this adaptation is the error between the exact surface  $\Gamma$  and its approximation, defined by the 0 isovalue of the SDF  $\Gamma_h$ . This error can be evaluated by considering the *Hausdorff distance* between those two surfaces. The Hausdorff distance is defined as follows :

**Definition 2.5.** *Hausdorff distance*

Two submanifolds  $\Omega_1$  and  $\Omega_2$  in  $\mathbb{R}^d$  are considered.

Denoting by  $d(x, \Omega_1) = \inf_{y \in \Omega_1} d(x, y)$  the euclidean distance from  $x$  to  $\Omega_1$  and defining  $\tilde{d}(\Omega_1, \Omega_2) = \sup_{x \in \Omega_1} d(x, \Omega_2)$ , the Hausdorff distance writes :

$$d^H(\Omega_1, \Omega_2) = \max(\tilde{d}(\Omega_1, \Omega_2), \tilde{d}(\Omega_2, \Omega_1))$$



Figure 2.6 – Area of 0 level set metric imposition

As proposed in [65], this error can be bounded using the curvature  $\kappa$  of the approximated surface with the following theorem (written for 3D problems) :

**Theorem 2.2. Upper bound of the Hausdorff distance**

Considering  $\Gamma_h$  the approximation of the exact surface  $\Gamma$  on the mesh  $\mathcal{T}^h$ . By denoting  $\mathcal{E}$  the set of elements  $T$  in  $\mathcal{T}^h$  intersecting the the 0 level set (i.e. such that the sign of  $u$  changes on the elements), we have :

$$d^H(\Gamma_h, \Gamma) \leq \max_{T \in \mathcal{E}} \frac{l_1^2 \kappa_1}{1 - h\kappa_1 - h^2 \kappa_1^2} + \frac{l_2^2 \kappa_2}{1 - h\kappa_2 - h^2 \kappa_2^2} \quad (2.41)$$

where  $h$  is the size of the element  $T$  in the normal direction to the surface and  $l_1, l_2$  are the sizes in the direction of the principal curvatures.  $\kappa_1$  and  $\kappa_2$  are the minimum and maximum values of the two local principal curvatures in  $T$ .

From this inequality, a metric field at each vertex in the elements of  $\mathcal{E}$  can be defined. Denoting the required error approximation by  $\epsilon$ , the metric proposed reads :

$$\mathcal{M} = \mathcal{R}^T \begin{pmatrix} \frac{1}{h_{min}^2} & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix} \mathcal{R} \quad (2.42)$$

where  $\mathcal{R} = (\nabla\psi_0 \mathbf{t}_1 \mathbf{t}_2)$ ,  $\psi_0$  being the 0 level set and  $(\mathbf{t}_1, \mathbf{t}_2)$  a basis of its tangential plan and

$$\lambda = \min \left( \max \left( \frac{|\kappa|}{\epsilon}, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right), \quad \kappa = \Delta\psi_0 \quad (2.43)$$

**Remark 2.1.** The metrics (2.42) are *a priori* defined for the nodes of elements intersecting the 0 level set. In practice, as shown figure 2.6, this metric is imposed in an area  $w$  around the surface, chosen according to the considered case, so as to produce a mesh suitable for Navier-Stokes simulations.

### 2.2.3.4 Metric intersection

The two metric fields defined for the SDF and the physical adaptation need to be combined so as to generate a mesh adapted to both criteria. To this end, we use *metric reduction* which is briefly described here. Denoting by  $\mathcal{M}_1$  and

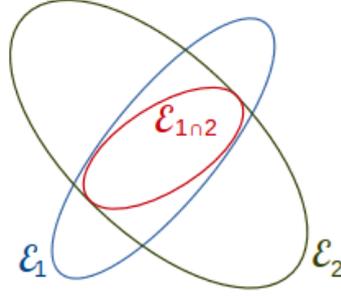


Figure 2.7 – Metric Reduction

$\mathcal{M}_2$  two metrics defined at the same node, the metric  $\mathcal{M}_{1\cap 2}$  resulting from the intersection must be such that both sizes prescribed by the two initial metrics are respected. From a geometrical point of view, as illustrated figure 2.7, the ellipsoid  $\mathcal{E}_{1\cap 2}$  associated to  $\mathcal{M}_{1\cap 2}$  is the biggest ellipsoid included in the ones associated to  $\mathcal{M}_1$  and  $\mathcal{M}_2$ ,  $\mathcal{E}_1$  and  $\mathcal{E}_2$ .

The process of the reduction, presented in some of the above references (see [14] for instance) is recalled here. Let define  $\mathcal{N} = \mathcal{M}_1^{-1}\mathcal{M}_2$ .  $\mathcal{N}$  is diagonalizable, and the normalized eigenvectors ( $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ ) provide a basis such that  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are congruent to a diagonal matrix in this basis. Defining :

$$\begin{aligned}
 \lambda_{i,1} &= \mathbf{e}_i^T \mathcal{M}_1 \mathbf{e}_i, & \lambda_{i,2} &= \mathbf{e}_i^T \mathcal{M}_2 \mathbf{e}_i, & i &= 1, \dots, 3 \\
 \mathcal{P} &= (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)
 \end{aligned}$$

The intersected metric is of the form :

$$\mathcal{M}_{1\cap 2} = \mathcal{P}^{-1T} \begin{pmatrix} \max(\lambda_{1,1}, \lambda_{1,2}) & 0 & 0 \\ 0 & \max(\lambda_{2,1}, \lambda_{2,2}) & 0 \\ 0 & 0 & \max(\lambda_{3,1}, \lambda_{3,2}) \end{pmatrix} \mathcal{P}^{-1}$$

### 2.2.4 Algorithm

To obtain a mesh optimised for a certain problem, the coupling CFD-remeshing algorithm 1 is employed. The iterations allow to obtain the convergence for the mesh and for the CFD. In this algorithm 1,  $k_{max}$  proposes to stop the iterative procedure if the solution is considered accurate enough, even if the process of CFD-remeshing coupling is not converged. Figures 2.8 and 2.9 presents the process for the adaptation to the SDF of a circle. On figure 2.8, the initial uniform mesh and the final adapted mesh are displayed, and on figure 2.9 the mesh obtained at the different iterations of the process.

For the penalized simulation that will be presented section 2.3.5, the initial mesh for CFD computation is adapted to the 0 level set and the iterations are performed to increase the solution accuracy.

---

**Algorithm 1** CFD-Remeshing convergence

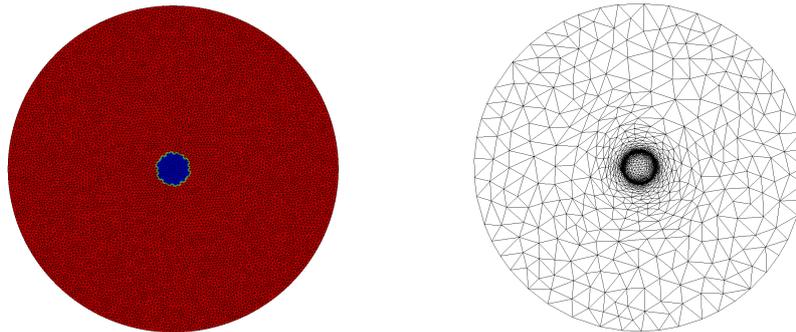
---

```

initial non adapted mesh  $\mathcal{T}_h^0$ 
 $\mathcal{T}_h^k = \mathcal{T}_h^0$ 
initial solution :  $u_h^0 = u_0, k = 0$ 
while ( $k < k_{max}$ ) do
    Compute  $u_h^{k+1}$ , converged solution of the considered problem on  $\mathcal{T}_h^k$ 
    Calculation of a metric field  $\mathcal{M}^{k+1}$ , associated to an error estimator
    defined on  $u_h^{k+1}$ 
    Generate a new mesh  $\mathcal{T}_h^{k+1}$  in agreement with  $\mathcal{M}^{k+1}$ 
    if (Convergence :  $\mathcal{T}_h^{k+1} \sim \mathcal{T}_h^k$ ) then
        Exit
    end if
    Compute  $\tilde{u}_h^{k+1}$ , interpolation of  $u_h^{k+1}$  on  $\mathcal{T}_h^{k+1}$ 
     $u_h^0 = \tilde{u}_h^{k+1}$ 
end while

```

---



(a) Initial mesh and level set (10,709 vertices) (b) Final mesh and level set isovalue(3,454 vertices)

Figure 2.8 – Mesh Adaptation to a circle

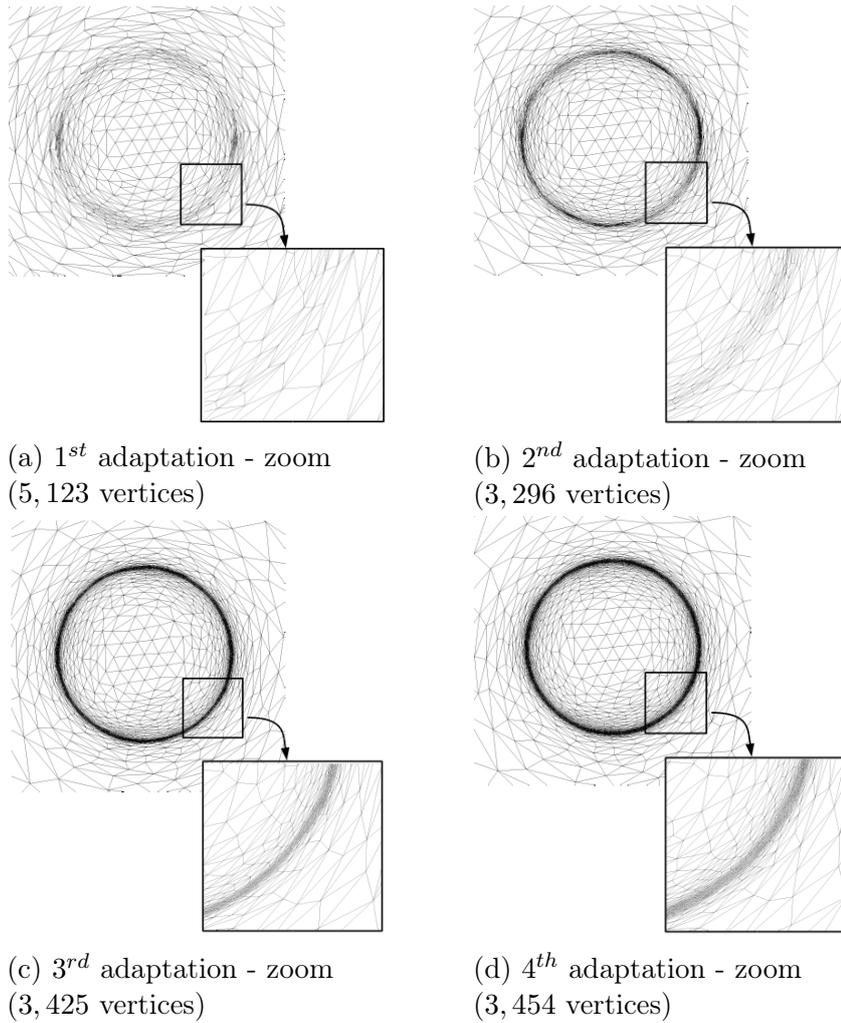


Figure 2.9 – Mesh/Solution convergence - four iterations

## 2.3 Discretization with Residual Schemes

In this section, the numerical approximation of steady problem is discussed. First, a general state of the art concerning Residual Distribution schemes is given. Then, we recall how to construct RD schemes for a scalar advection diffusion equation. Finally, we get to the main contribution of this study : the extension of the residual distribution framework to the penalized Navier Stokes equations. The potential of the proposed approach is evaluated on several test cases.

### 2.3.1 State of the art

Three main categories of schemes can be employed for the numerical resolution on unstructured grids. The Finite Volume (FV) method [108, 24, 23, 149] evolves cell averaged values of the solution. First order schemes are quite easy to build, the solution being approximated by a constant on each cell. Higher order schemes however, require to reconstruct polynomial solutions using constant data from neighbouring cells. Limitation techniques are employed to prevent oscillations, leading *e.g.* to ENO/WENO schemes [149]. The main drawback of such method is the increase of the stencil with the order, leading to complex implementation especially on distributed memory parallel machines. Another possible approach is the finite element (FE) method. Classical Galerkin formulation being unstable for advection problem, several techniques have been studied to provide stabilization such as the well known SUPG [35], or Galerkin/Least Square method [90] (see also [92] or [91] for reviews on those techniques). The construction of high order schemes with those approaches is easy, as the accuracy of the solution depends on the order of the local polynomial basis functions. This locality allows in addition an efficient parallelization, as the stencil remains compact while the order of accuracy increase. The price to pay is an increase in storage, and number of degree of freedom. When dealing with solutions involving shocks, additional *shock capturing* terms are required to keep stable schemes [92]. The last kind of method is the Discontinuous Galerkin (DG) method, introduced by Reed and Hill [132] (see also the works proposed by Cockburn, Shu and co authors [46, 45, 47]). DG method combines the FV and FE framework. It is based on a Galerkin formulation, but no continuity is sought between the elements. Thus, numerical fluxes are employed to handle the discontinuities, as performed in FV schemes. These schemes also have very compact stencil and are thus very easy to parallelize. The main drawback of such method are the definition of high order preserving limiters, to avoid oscillations in shocks, while preserving the high accuracy, and the large number of DoF, even compared to stabilized continuous FE[93].

In this work we consider another alternative, the Residual Distribution

(RD) schemes, introduced with the notion of fluctuation splitting by Roe in 1982 [140]. These schemes share many properties with FE, and their advantages, such as compact stencil and an easy increase of the accuracy order. Since then, many works on the subject have been proposed, especially by Roe, Deconinck, Abgrall, Ricchiuto and their coauthors. Their contributions are part of an exhaustive list in Ricchiuto's HDR [134]. The PhDs of Caranei (2000) [41], Dobeš (2007) [60], Larat (2009) [106], Villedieu (2009) [160], and more recently Sermeus (2013) [147] and De Santis (2013) [56] propose also a good basis, especially concerning the treatment of the viscous flux in advection diffusion problems. We recall here some of the contributions in this field of research. The construction of such schemes for advection equations is proposed by Struijs and co authors [153, 154], Van der Weide and Deconinck [58], Abgrall [1], Csík *et. al.* [53], and very high order ( $> 2$ ) can be found in [6] or [86, 88]. A formulation of the Lax Wendroff theorem is defined by de Mer and Abgrall in [7] and a systemic construction of monotone high order schemes is proposed in [9]. A publication dedicated to the comparison against FV schemes is proposed by Guzik and Groth [80].

Now, focusing on advection diffusion resolution, high order schemes have been proposed in [161, 136] where diffusion is treated with a Galerkin discretization. Caranei and Fuchs proposed  $2^{nd}$  and  $3^{rd}$  order scheme to solve the Navier Stokes equations and perform LES simulations [41, 39, 40]. In De Santis PhD, inspired from the work of Nishikawa [122] (see also recent results in [115], and [114] for  $1D$  problems solved with  $6^{th}$  order schemes), writing the problem as a first order system with the gradient considered as a new unknown, high order schemes can be constructed [10, 56]. This is the method employed in this work.

### 2.3.2 Notations for mesh and approximate solution

Let  $\mathcal{T}^h$  denote the mesh discretization of the spatial domain  $\Omega$ . The generic triangle (tetrahedron) of  $\mathcal{T}^h$  is denoted by  $T$ . We denote by  $\partial T$  the element boundary, and by  $|T|$  its area (volume in  $3D$ ). By  $\partial T_i$  we denote the face of  $\partial T$  opposite to node  $i$ , and  $\mathbf{n}_i$  denotes the inward normal scaled by  $|\partial T_i|$  (see figure 2.10a). The following notation will also be used throughout this manuscript : the set of triangles containing node  $i$  is denoted by  $\{T \ni i\}$ . Thus, we define  $\sum_{T \ni i}$  for a summation on the elements in the neighbourhood of a node and  $\sum_{T \ni (i,j)}$  for a summation on the elements an edge belongs to. The dual cell associated to a node  $i$  (see figure 2.10b) is denoted by  $C_i$ , of area  $|C_i|$ .

As for finite element method, the approximated solution is defined as  $u_h(\mathbf{x}, t) = \sum_{i=1}^{N_{Dof}} u_i(t) \varphi_i(\mathbf{x})$ . With  $N_{Dof}$  we denote the number of degrees of freedom of the element, and  $\varphi_i$  the basis function associated to node  $i$ , with  $u_i$  the value of the solution at the node. In this work, the degrees of freedom are the nodes of the elements and the basis functions are the  $P^1$  Lagrange

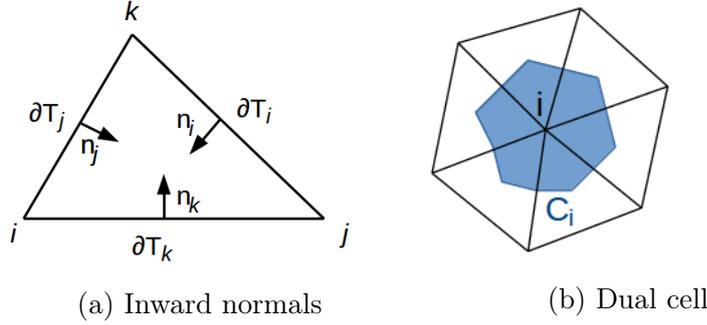


Figure 2.10 – Notations for triangles

functions. Some properties that will be useful for the following are :

$$\int_T f(\mathbf{x}) = \sum_{j=1}^3 \frac{f_j}{d+1}, \quad \nabla \varphi_i = \frac{\mathbf{n}_i}{d|T|} \quad (2.44)$$

where  $d$  denotes the number of dimension.

### 2.3.3 Steady scalar conservation laws

As said in the introduction of this chapter, to acquire general notions on RD schemes, the construction is performed for a 2D scalar conservation law.

#### 2.3.3.1 Principle and construction

We consider the steady case of equation (2.30) :

$$\nabla \cdot \mathcal{F}(u) = 0 \quad \text{on } \Omega \quad (2.45)$$

By integrating equation (2.45) evaluated using the approximated solution on each  $T \in \mathcal{T}^h$ , the *element fluctuation* is defined as :

$$\phi^T(u_h) = \int_T \nabla \cdot \mathcal{F}(u_h) \quad (2.46)$$

**Remark 2.2.** What has been called here *total fluctuation* can be referred in the literature as *total residual*. In this manuscript, the notion of residual will be used for unsteady problems.

To account for the BC, one can include *face fluctuations* defined by :

$$\phi^f(u_h) = \int_f (\mathcal{F}^{BC}(u_h, u_{BC}, \mathbf{n}) - \mathcal{F}(u_h) \cdot \mathbf{n}) \quad (2.47)$$

where  $\mathcal{F}^{BC}(u_h, u_{BC}, \mathbf{n})$  is the numerical flux consistent with the BC dependent of the exact or physical value of the solution available on the boundary  $u_{BC}$ , the local state  $u_h$  and the boundary normal  $\mathbf{n}$ .

Once the fluctuations are defined on each element, the idea is to distribute them to the degrees of freedom into *nodal fluctuation*  $\phi_i^T(u_h)$ . It can generically be written using *distribution coefficient*  $\beta_i^T$  and :

$$\phi_i^T(u_h) = \beta_i^T \phi^T(u_h) \quad (2.48)$$

An explicit definition of these distribution coefficients is not necessarily provided, and often only the expression of  $\phi_i^T$  is used.

This distribution process is the heart of RD schemes and depends on the spatial discretization. More details will be given in section 2.3.3.4. In the same way, face fluctuations are split to the degree of freedom of the face, defining nodal face fluctuation  $\phi_i^f$ . The next step leading to the final form of a RD scheme is to gather the information coming from all the elements a node belongs to. The summation of all the contributions must be null :

$$\sum_{T \ni i} \phi_i^T(u_h) + \sum_{(f \in \partial T_h) \ni i} \phi_i^f(u_h) = 0 \quad (2.49)$$

One possibility to define the nodal face fluctuation is :

$$\phi_i^T(u_h) = \int_T \varphi_i(\mathcal{F}^{BC} - \mathcal{F} \cdot \mathbf{n}) \quad (2.50)$$

that satisfies the conservation :  $\sum_{i \in \partial T} \phi_i^f = \phi^f$ , as proposed in De Santis'PhD [56]. For far field boundaries, this is the approach used here. As this will not be furthered developed, we will omit this term to simplify the equations.

The whole process for the construction is sketched on figure 2.11. To solve this problem (2.49), the following pseudo unsteady problem is considered :

$$\begin{cases} \frac{\partial u_h}{\partial t} + \sum_{T \ni i} \phi_i^T(u_h) = 0 \\ u_h(t = 0, \mathbf{x}) = u_0(\mathbf{x}) \end{cases} \quad (2.51)$$

Where  $u_0$  is the initial solution.

For the temporal discretization, an Euler scheme is employed even if more sophisticated ones could be used. It leads to the following iterative procedure :

$$\frac{|C_i|}{\Delta t^n} (u_i^{n+1} - u_i^n) + \sum_{T \ni i} \phi_i^T(u_h^*) = 0 \quad (2.52)$$

Where  $\Delta t^n$  is the pseudo time step. The notation  $*$  is employed to let the possibility of an implicit or explicit scheme. For the steady problems, the

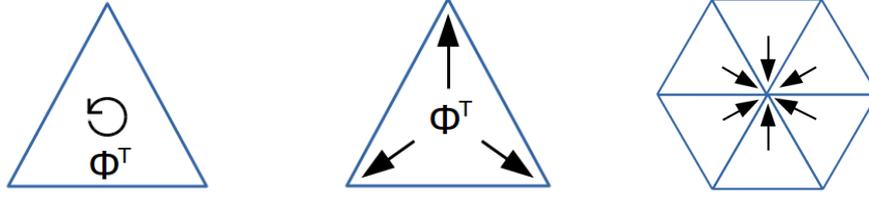


Figure 2.11 – RD process. Left : fluctuation computation - Middle : fluctuation distribution - Right : Nodal fluctuation gathering

developments proposed by De Santis during his PhD [56] are used. An Euler implicit scheme is employed and solved using a Newton algorithm. Denoting  $R(u_h) = \sum_{T \ni i} \phi_i^T(u_h)$  and  $J = \partial R(u)/\partial u$  its Jacobian, the problem (2.52) writes :

$$\left( \frac{\mathbb{I}}{\Delta t^n} + J(u_h^n) \right) (u_h^{n+1} - u_h^n) = -R(u_h^n) \quad (2.53)$$

Different algorithms have been developed in [56] to which we refer the reader for more details. An extension to time dependant problem has been performed section 3.1.5.2 for the development of the Crank-Nicolson scheme. Although more approximation on the Jacobian are proposed, the principle is identical.

The computation of the element residual is performed by integrating the flux on the boundary  $\partial T$  :

$$\phi^T(u_h) = \int_{\partial T} \mathcal{F}(u_h) \cdot \hat{\mathbf{n}}^{out} = \sum_{j \in T} \int_{\partial T_j} \mathcal{F}(u_h) \cdot \hat{\mathbf{n}}_j^{out} \quad (2.54)$$

where  $\hat{\mathbf{n}}^{out}$  denotes the outward normal.

The quasi linear form of the equation can also be introduced to compute the element fluctuation :

$$\phi^T(u_h) = \int_T \nabla \cdot \mathcal{F}(u_h) = \bar{\mathbf{a}} \cdot \int_T \nabla u_h = \sum_{i \in T} \frac{1}{d} \bar{\mathbf{a}} \cdot \mathbf{n}_i u_i \quad (2.55)$$

with  $\bar{\mathbf{a}}$  obtained from the *exact evaluation* of

$$\bar{\mathbf{a}} = \frac{1}{|T|} \int_T \mathbf{a} \quad (2.56)$$

and having introduced *inflow parameter*  $k_i$  :

$$k_i = \frac{1}{d} \bar{\mathbf{a}} \cdot \mathbf{n}_i \quad (2.57)$$

it is obtained :

$$\phi^T(u_h) = \sum_{i \in T} k_i u_i \quad (2.58)$$

However, such an approach cannot be considered, unless smooth solutions are studied, owing to the cost necessary to evaluate  $\bar{\mathbf{a}}$  such that the resulting scheme is conservative [12]. As we will see in the following, the inflow parameter (2.57) can still be used to define distribution coefficients.

### 2.3.3.2 Basic Properties

In this section, the properties and theorem are recalled without demonstration. For more details, we refer the reader to the references reported in the text. The starting point is the definition of conservation and consistency with a Lax-Wendroff like theorem. Then, the properties ensuring a maximum principle and monotonicity of RD schemes are given. Finally, the high order accuracy is defined and Godunov's theorem is expressed in the RD framework.

#### Consistency and convergence

Let begin with the notion of conservation and consistency. Conservation of the quantity requires classical hypothesis concerning the flux and solution approximation at the edges of the elements.

##### Property 2.1. Conservation

By defining the fluctuation (2.46) such that for two neighbours elements  $T_1$  and  $T_2$  of boundary  $\Gamma$ , the flux evaluated at the approximated solution satisfies :

$$\mathcal{F}(u_h^{T_1}) \cdot \mathbf{n}_\Gamma^{T_1} = -\mathcal{F}(u_h^{T_2}) \cdot \mathbf{n}_\Gamma^{T_2} \quad (2.59)$$

and with nodal fluctuation satisfying

$$\sum_{i \in T} \phi_i^T(u_h) = \phi^T(u_h) \quad (2.60)$$

then the scheme (2.49) is conservative.

Now, considering that the nodal fluctuation can be written over the form (2.48), then the notion of consistency can be imposed on the distribution coefficients :

##### Property 2.2. Consistency

If there exist coefficients  $\beta_i^T$  such that we can explicitly write the scheme over the form (2.48), then :

$$\sum_{j \in T} \beta_j^T = 1 \quad (2.61)$$

Now, considering a conservative RD scheme, the Lax-Wendroff theorem, ensuring the convergence to a weak solution of the problem is expressed as follows [7, 9, 106, 56, 8] :

**Theorem 2.3. Lax-Wendroff like**

Let consider a sequence of approximated solutions  $(u_h)_h$  for given meshes  $\mathcal{T}_h$ , solution of (2.49). Assuming that there exists  $u \in \mathcal{L}^2(\mathbb{R}^2)$  and a constant  $C$  depending only on the mesh such that :

$$\sup_h \sup_{\mathbf{x}} |u_h(\mathbf{x})| \leq C, \quad \lim_{h \rightarrow 0} \|u - u_h\|_{\mathcal{L}^2_{loc}(\mathbb{R}^2)} = 0 \quad (2.62)$$

Then  $u$  is a weak solution of the problem (2.45).

The proof of this theorem is provided in [7, 9].

**Local Positivity and Monotonicity Preserving Property**

Numerical instabilities across discontinuities have always been an issue in CFD. Solutions of conservation laws might lack of regularity or even develop discontinuities. Instabilities or blow up may be observed numerically. The aim is then to design schemes that will verify a *discrete maximum principle* defined as follows :

$$\forall i \in \mathcal{T}^h, \forall n, \min_{j \in \mathcal{T}^h | ij \in \mathcal{T}} u_j^n \leq u_i^{n+1} \leq \max_{j \in \mathcal{T}^h | ij \in \mathcal{T}} u_j^n \quad (2.63)$$

This property ensures a  $\mathcal{L}^\infty$  stability. In previous works ([106, 71, 56, 147] among others), some propositions recalled here ensure a non oscillatory character for RD schemes. First, the nodal fluctuation (2.48) must be rewritten in the form :

$$\phi_i^T(u_h) = \sum_{\substack{j \in \mathcal{T}^h \\ j \neq i}} c_{ij}(u_i - u_j) \quad (2.64)$$

The *local positivity* can then be defined by :

**Property 2.3. Local Positivity**

A RD scheme is said locally positive if the nodal residuals defined with (2.64) are such that :

$$\forall T \in \mathcal{T}^h, \forall (i, j)_{i \neq j} \in T, \quad c_{ij} \geq 0 \quad (2.65)$$

This local positivity property has relations with the Total Variation Diminishing (TVD) theory settled for high order FV schemes [149]. From there, the *Monotonicity Preservation* (MP) property, that ensures the non oscillatory character of a RD scheme, can be expressed as :

**Property 2.4. Monotonicity Preservation**

If a scheme expressed over the formulation (2.64) satisfies the local positivity property 2.3 and (in the case of an explicit discretization in time) the CFL constraint :

$$\frac{\Delta t}{|C_i|} \sum_{T \ni i} \sum_{j \in T} c_{ij} \leq 1 \tag{2.66}$$

the discrete solution satisfies the discrete maximum principle 2.63.

**Accuracy and Godunov Theorem**

The question of accuracy is now characterized. The definition of accuracy for RD schemes has been addressed in [106, 56, 134, 8] :

**Definition 2.6. Accuracy of RD scheme**

Let  $\psi$  be a smooth function  $\psi \in C_0^{r+1}(\Omega)$  and  $\psi_h$  its  $r^{\text{th}}$  accurate approximation on a mesh  $\mathcal{T}^h$  discretization of  $\Omega$ . Considering  $u \in H^{r+1}$  an exact smooth function satisfying (2.45), and  $u_h$  its  $r^{\text{th}}$  polynomial approximation of degree  $r$ . Defining the following truncation error :

$$\epsilon(u_h, \psi) = \sum_{i \in \mathcal{T}^h} \psi_i \sum_{T \ni i} \phi_i^T(u_h) \tag{2.67}$$

a RD scheme is said  $(r + 1)^{\text{th}}$  order accurate if :

$$|\epsilon(u_h, \psi)| \leq \mathcal{O}(h^{r+1}) \tag{2.68}$$

and the following property has been demonstrated (see [9, 1]) :

**Property 2.5. High Order Accuracy**

A RD scheme defined as (2.49) is  $(r + 1)^{\text{th}}$  accurate in the sense of definition 2.6 provided that the resulting approximated solution satisfies for a  $d$  dimensional case :

$$\sup_{T \in \mathcal{T}^h} \sup_{i \in T} |\phi_i^T(u_h)| = \mathcal{O}(h^{r+d}) \tag{2.69}$$

The following property called *Linearity Preservation* (LP) property proposes a simple condition to achieve high order accuracy :

**Property 2.6. Linearity Preservation**

A RD scheme is said linearity preserving if its distribution coefficients  $\beta_i^T$  satisfying the consistency property (2.2) are uniformly bounded :

$$\max_{T \in \mathcal{T}^h} \max_{i \in T} |\beta_i^T| < \infty \tag{2.70}$$

It is easily seen that a scheme satisfying the LP property 2.6 satisfies the required hypothesis for the accuracy property 2.5 and thus lead to a high order scheme.

Now we are willing to formulate the classical Godunov theorem in the RD framework. First, let give a definition of a linear scheme :

**Definition 2.7. Linear Scheme**

A RD scheme whom nodal fluctuation is expressed over the formulation (2.64) is said linear if the coefficients  $c_{ij}$  does not depend on the discrete solution of the problem.

This definition of linear scheme allows to express the classical Godunov theorem in the RD framework.

**Theorem 2.4. Godunov**

A RD scheme being at the same time monotone in the sense of property 2.4 and satisfying linearity preserving condition cannot be linear.

Non linear schemes are required to combine high order of accuracy with the discrete maximum principle.

**2.3.3.3 Analogy with Stabilized Finite Elements**

One interesting RD scheme property that will be useful in the following is their analogy with stabilized finite elements [147, 126]. Indeed, by considering a test function  $\omega_i$  as :

$$\omega_i = \varphi_i + \gamma_i \tag{2.71}$$

corresponding to the basis function plus a perturbation  $\gamma_i$  (also called bubble function), the problem (2.45) can be discretized over the following stabilized finite element (or Petrov Galerkin (PG)) formulation :

$$\sum_{T \in \mathcal{T}^h} \int_T \omega_i \nabla \cdot \mathcal{F}(u_h) = 0 \tag{2.72}$$

and considering  $P^1$  case, we can easily see that if we set :

$$\gamma_i = \beta_i^T - \frac{1}{N_{DoF}} \tag{2.73}$$

then

$$\phi_i^T = \int_T \omega_i \nabla \cdot \mathcal{F}(u) = \beta_i^T \phi^T \tag{2.74}$$

This analogy will be useful, in particular for the extension to unsteady problems.

**Remark 2.3.** The consistency property 2.2 can also be expressed on  $\omega_i$  as :

$$\sum_{i \in T} \omega_i = 1 \tag{2.75}$$

### 2.3.3.4 Two examples of RD schemes

For the studies performed in this PhD, two schemes are used and presented here : the *SUPG* and *Limited Lax Friedrich* schemes. One can find other example in [56, 60, 106, 147, 134].

#### SUPG scheme

The SUPG scheme is exactly the RD form of the well known stabilized FE method [35]. It is an equirepartition of the total fluctuation (centered part) to the degrees of freedom plus the classical streamline diffusion term. Thus, it gives the following nodal fluctuations :

$$\phi_i^T(u_h^n) = \frac{\phi^T(u_h)}{N_{DoF}} + \int_T (\mathbf{a} \cdot \nabla \varphi_i) \tau (\mathbf{a} \cdot \nabla u_h) \quad (2.76)$$

Using relation (2.44) and the definition of the inflow parameter (2.57), it leads to the following distribution coefficients :

$$\beta_i^T = \frac{1}{N_{DoF}} + k_i \tau \quad (2.77)$$

where  $\tau$  is the stabilization parameter, where in the scalar case may be defined as :

$$\tau = \left( \sum_{j \in T} k_j^+ \right)^{-1}, \quad k_j^+ = \max(0, k_j) \quad (2.78)$$

This scheme is linear and thus does not satisfy the Godunov theorem 2.4. It will be used for problems having smooth solutions.

#### Limited Lax Friedrich Scheme

This scheme starts from a local Lax Friedrich scheme, reading :

$$\phi_i^T(u_h) = \frac{\phi^T(u_h)}{N_{Dof}} + \alpha (u_i - \bar{u}), \quad \alpha \geq \max_{j \in T} (|k_j| + \nu) \quad (2.79)$$

This is a central scheme going along with the local Lax Friedrich stabilization suitable for discontinuities. However, being linear, it cannot verify the accuracy conditions 2.5/2.6. The distribution coefficients are defined as :

$$\beta_i^T = \frac{\phi_i^T(u_h)}{\phi^T(u_h)} \quad (2.80)$$

The  $\beta_i^T$  defined with (2.80) may grow unbounded as there is no reason that the term  $\frac{u_i - \bar{u}}{\phi^T}$  remains bounded when  $\phi^T \rightarrow 0$ . Thus, as said previously, a limitation technique denoted as *PSI limitation* is used, fully discussed in

[2, 6, 71], that consists into bounding the quantities into limited distribution coefficients  $\beta_i^*$ , computed as follows :

$$\beta_i^* = \frac{\max(\beta_i, 0)}{\sum_{j \in T} \max(\beta_j, 0)} \quad (2.81)$$

and the limited scheme writes :

$$\phi_i^{T,*}(u_h) = \beta_i^* \phi^T(u_h) \quad (2.82)$$

As deeply discussed in [6, 2, 138], the schemes obtained with this limiting strategy admit some spurious modes, which become more visible for smooth problems, hampering the obtaining of high order accuracy in practice. To correct this, in [6], it is proposed to add, only in smooth regions, a streamline dissipation term which allows to filter those spurious modes, while preserving the high order accuracy. This leads to the following expression of the *Limited Lax Friedrich* (LLF) scheme :

$$\phi_i^T(u_h) = \beta_i^{T,*} \phi^T(u_h) + \theta \int_T (\mathbf{a} \cdot \nabla \varphi_i) \tau(\mathbf{a} \cdot \nabla u_h) \quad (2.83)$$

where  $\theta$  is a shock detector such that  $\theta \approx 1$  in smooth area and  $\theta \ll 1$  across discontinuities.

### 2.3.3.5 Extension to advection/diffusion problems

As explained in the state of the art, the method employed to deal with advection diffusion problems is the one proposed by De Santis [56, 10]. With this approach, a high order discrete approximation of the gradient is obtained, even for cases which are not advective dominated. Considering the steady formulation of the advection diffusion equation (2.33) :

$$\nabla \cdot \mathcal{F}(u) = \nabla \cdot (\nu \nabla u) \quad (2.84)$$

the idea is to consider the gradient as a new unknown of the problem and to solve the first order system (FOS) :

$$\begin{cases} \nabla \cdot \mathcal{F}(u) - \nabla \cdot (\nu \mathbf{q}) = 0 \\ \mathbf{q} - \nabla u = 0 \end{cases} \quad (2.85)$$

Writing a SUPG type weak form for this system gives :

$$\int_T \varphi_i \begin{pmatrix} \nabla \cdot \mathcal{F}(u_h) - \nabla \cdot (\nu \mathbf{q}_h) \\ \mathbf{q}_h - \nabla u_h \end{pmatrix} + \int_T \underline{\mathbf{A}} \cdot \nabla \varphi_i \underline{\mathcal{T}} \begin{pmatrix} \nabla \cdot \mathcal{F}(u_h) - \nabla \cdot (\nu \mathbf{q}_h) \\ \mathbf{q}_h - \nabla u_h \end{pmatrix} = 0 \quad (2.86)$$

with :

$$\underline{\mathbf{A}} \cdot \nabla \varphi_i = \begin{pmatrix} \mathbf{a} \cdot \nabla \varphi_i & -\nu \frac{\partial \varphi_i}{\partial x} & -\nu \frac{\partial \varphi_i}{\partial y} \\ -\frac{\partial \varphi_i}{\partial x} & 0 & 0 \\ -\frac{\partial \varphi_i}{\partial y} & 0 & 0 \end{pmatrix} \quad (2.87)$$

and  $\underline{\boldsymbol{\tau}}$  is assumed of the form :

$$\underline{\boldsymbol{\tau}} = \begin{pmatrix} \tau_a & 0 & 0 \\ 0 & \tau_d & 0 \\ 0 & 0 & \tau_d \end{pmatrix} \quad (2.88)$$

where  $\tau_a$  and  $\tau_d$  are positive coefficients.

Considering a reconstructed gradient at each DoF in the system (2.85)  $\mathbf{q} = \widetilde{\nabla} u_h$ , the first equations resulting from (2.86) reads :

$$\begin{aligned} \int_T \varphi_i \nabla \cdot \left( \mathcal{F}(u_h) - \nabla \cdot (\nu \widetilde{\nabla} u_h) \right) + \int_T (\mathbf{a} \cdot \nabla \varphi_i) \tau_d \nabla \cdot \left( \mathcal{F}(u_h) - \nu \widetilde{\nabla} u_h \right) \\ + \int_T \nu \nabla \varphi_i \cdot \left( \tau_d \left( \nabla u_h - \widetilde{\nabla} u_h \right) \right) = 0 \end{aligned} \quad (2.89)$$

From this formulation, De Santis proposed to modify the schemes presented previous section by adding a term penalizing the difference between the the discontinuous gradient (computed with the basis function) and the reconstructed one  $\widetilde{\nabla} u_h$  [56, 10], inspired from the last integral of the equation (2.89). The element fluctuation is thus computed as :

$$\phi^T(u_h) = \int_T \left( \nabla \cdot \mathcal{F}(u_h) - \nabla \cdot (\nu \widetilde{\nabla} u_h) \right) \quad (2.90)$$

the new form of the schemes writes :

$$\phi_i^{SUPG} = \beta_i^{T,SUPG} \phi^T(u_h) + \int_T \nu \nabla \varphi_i \cdot \left( \nabla u_h - \widetilde{\nabla} u_h \right) \quad (2.91)$$

$$\begin{aligned} \phi_i^{LLF} = \beta_i^* \phi^T(u_h) + \theta \int_T (\mathbf{a} \cdot \nabla \varphi_i - \nabla \cdot (\nu \nabla \varphi_i)) \tau (\mathbf{a} \cdot \nabla u_h) \\ + \int_T \nu \nabla \varphi_i \cdot \left( \nabla u_h - \widetilde{\nabla} u_h \right) \end{aligned} \quad (2.92)$$

Numerical results can be found in the above references with different gradient recovery strategies. In the scope of this PhD, the adopted one is the area-weighted method :

$$\widetilde{\nabla} u_i = \frac{\sum_{T \ni i} \nabla u_h |T|}{\sum_{T \ni i} |T|} \quad (2.93)$$

From a computational point of view, the advective part is computed with (2.54) and the diffusive one by passing through the edges :

$$\int_T \nabla \cdot (\nu \widetilde{\nabla} u_h) = \sum_{i \in T} \int_{\partial T_i} \nu \widetilde{\nabla} u_h \cdot \widehat{\mathbf{n}}_i^{out} \quad (2.94)$$

### 2.3.4 Extension to the penalized Navier Stokes system

#### 2.3.4.1 RD schemes for the Navier Stokes equations

The schemes presented previously have been used for the NS equations in [56]. Using the notations introduced in section 2.1.1, the fluctuation is computed as :

$$\phi^T(\mathbf{u}_h) = \int_T \left( \nabla \cdot \mathbf{F}^{Eul}(\mathbf{u}_h) - \nabla \cdot \left( \mathbb{K} \widetilde{\nabla \mathbf{u}_h} \right) \right) \quad (2.95)$$

and is computed by integrating the flux on the boundary as :

$$\phi^T(\mathbf{u}_h) = \sum_{i \in T} \int_{\partial T_i} \left( \mathbf{F}^{Eul}(\mathbf{u}_h) - \mathbb{K} \widetilde{\nabla \mathbf{u}_h} \right) \cdot \hat{\mathbf{n}}_i \quad (2.96)$$

Similarly to the scalar case, inflow matrices, extension of the inflow parameter (2.57), are defined by :

$$\underline{\mathbf{K}}_i = \frac{1}{d} \sum_{k=1}^d \mathbb{A}_k(\mathbf{u}_h) n_{i_k} \quad (2.97)$$

#### SUPG Scheme

The SUPG scheme applied to the NS equations reads :

$$\begin{aligned} \phi_i^T(\mathbf{u}_h) = & \frac{\phi^T(\mathbf{u}_h)}{N_{DoF}} + \int_T \left[ (\mathbb{A} \cdot \nabla \varphi_i) \underline{\boldsymbol{\tau}} \left( \mathbb{A} \cdot \nabla \mathbf{u}_h - \nabla \left( \mathbb{K} \widetilde{\nabla \mathbf{u}_h} \right) \right) \right] + \\ & \int_T (\mathbb{K} \nabla \varphi_i) \cdot \left( \nabla \mathbf{u}_h - \widetilde{\nabla \mathbf{u}_h} \right) \end{aligned} \quad (2.98)$$

where  $\underline{\boldsymbol{\tau}}$  is now a stabilization matrix :

$$\underline{\boldsymbol{\tau}} = \frac{|T|}{N_{DoF}} (\mathbb{A}^+ + \mathbb{K})^{-1} \quad (2.99)$$

with  $\mathbb{A}^+ = \sum_{i \in T} \underline{\mathbf{R}}_{n_i} \underline{\boldsymbol{\Lambda}} \mathbf{L}_{n_i}$ ,  $\underline{\mathbf{L}}_n$  and  $\underline{\mathbf{R}}_n$  correspond respectively to the matrices of the left and right eigenvectors of the Euler Jacobian  $\mathbb{A}$  along the direction of the normal  $\mathbf{n}$ , and  $\underline{\boldsymbol{\Lambda}}$  denotes the corresponding eigenvalues matrix. The operator  $(\cdot)^+$  sets the negative values to 0.

The following distribution matrices  $\underline{\boldsymbol{\beta}}_i^T$  can be defined :

$$\underline{\boldsymbol{\beta}}_i^T = \frac{1}{N_{DoF}} \underline{\mathbf{I}} + \underline{\mathbf{K}}_i \underline{\boldsymbol{\tau}} \quad (2.100)$$

so as to rewrite the scheme over the form :

$$\phi_i^T(\mathbf{u}_h) = \underline{\boldsymbol{\beta}}_i^T \phi^T(\mathbf{u}_h) \quad (2.101)$$

### LLF scheme

Starting again from the local LF scheme :

$$\phi_i^T(\mathbf{u}_h) = \frac{\phi^T(\mathbf{u}_h)}{N_{DoF}} + \frac{1}{N_{DoF}} \alpha(\mathbf{u}_i - \bar{\mathbf{u}}) \quad (2.102)$$

a limitation procedure is required. As proposed in [56], the mapping for a system of equation is done in the characteristic space. Thus, first, the vector of total fluctuations and nodal fluctuations are rewritten :

$$\psi_i^T = \underline{\mathbf{L}}_n \phi_i^T, \quad \psi^T = \sum_{i \in T} \psi_i^T \quad (2.103)$$

where the mean fluid velocity vector gives the normal direction for the computation of the eigenvectors. As previously, for each variable *var* of the problem, unbounded distribution coefficients are defined :

$$\beta_i^{T,var} = \frac{\psi_i^{T,var}}{\psi^{T,var}} \quad (2.104)$$

and the PSI limiter (2.81) is applied to compute the limited distribution coefficients  $\beta_i^{T,*var}$ . Considering the distribution matrix  $\underline{\beta}_i^{T,*}$  diagonal with the corresponding limited coefficient in the diagonal, the nodal vector of fluctuation is, in the characteristic space :

$$\psi_i^{T,*} = \underline{\beta}_i^{T,*} \psi^T \quad (2.105)$$

and the limited vector of nodal residual in the physical space reads :

$$\phi_i^{T,*} = \underline{\mathbf{R}}_n \psi_i^{T,*} \quad (2.106)$$

Finally, with the additional streamline diffusion, the LLF scheme is :

$$\begin{aligned} \phi_i^{T,LLF} = \phi_i^{T,*} + \theta \int_T \left[ (\mathbb{A} \cdot \nabla \varphi_i - \nabla \cdot (\mathbb{K} \nabla \varphi_i)) \underline{\boldsymbol{\tau}} \left( \mathbb{A} \cdot \nabla \mathbf{u}_h - \nabla \left( \mathbb{K} \widetilde{\nabla} \mathbf{u}_h \right) \right) \right] \\ + \int_T (\mathbb{K} \nabla \varphi_i) \cdot \left( \nabla \mathbf{u}_h - \widetilde{\nabla} \mathbf{u}_h \right) \end{aligned} \quad (2.107)$$

#### 2.3.4.2 Modification of the schemes for the Penalization

Let recall the considered matrix system :

$$\nabla \cdot \mathbf{F}^{Eul}(\mathbf{u}) = \nabla \cdot (\mathbb{K} \nabla \mathbf{u}) + \mathbf{S} \quad (2.108)$$

Recalling that inside the solids, the source term is preponderant with respect to the other terms, it leads to a degeneration of the NS equations. Inside the

solid, what is solved is almost  $\mathbf{S} = \mathbf{0}$ . In this situation, considering the analogy with PG formulation presented section 2.3.3.3, a simple Galerkin discretization is sought. Thus, the discretization writes :

$$\sum_T \left\{ \int_T \omega_i (\mathbb{A} \cdot \nabla \mathbf{u}_h - \nabla \cdot (\mathbb{K} \nabla \mathbf{u}_h)) \right\} = \sum_{T \ni i} \left\{ \int_T \varphi_i \mathbf{S} \right\} \quad (2.109)$$

That corresponds in the RD framework to an equal distribution of the fluctuation associated to the source term, that will be denoted  $\phi^{T,S}$ . In the meantime, the fluctuation associated to the NS part  $\phi^{T,NS}$  is distributed using the distribution matrices associated to the chosen spatial scheme :

$$\begin{aligned} \sum_{T \ni i} \left\{ \frac{\beta_i^T}{N_{DoF}} \phi^{T,NS}(\mathbf{u}_h) + \frac{1}{N_{DoF}} \phi^{T,S}(\mathbf{u}_h) \right\} &= 0 \\ \phi^{T,NS}(\mathbf{u}_h) &= \int_T \left( \mathbb{A} \cdot \nabla \mathbf{u}_h - \nabla \cdot (\mathbb{K} \widetilde{\nabla} \mathbf{u}_h) \right) \\ \phi^{T,S} &= \int_T \mathbf{S} \end{aligned} \quad (2.110)$$

In addition, a tricky point needs to be considered that is the computation of the gradient reconstruction close to the interface. Indeed, looking at the different methods proposed in [56], the gradient is computed in each point using the neighbours. But for points on the boundary between fluid/structure, the penalized points simulating the body will add unphysical contributions (null for a motionless solid for instance). Looking at figure 2.12, if the simulation is performed without penalization (body fitted), the solid boundary is discretized with nodes  $N4, N, N7$  and only contributions from empty triangles are used for the reconstruction of the gradient at node  $N$ , so that only information in nodes  $N1, N2, N3, N4, N7$  is used. When using penalization, if no special treatment is done, additional contributions from the filled triangles will be added. These contributions are unphysical and result in a defective reconstruction. To overcome this difficulty, the area-weighted method proposed in [56] is modified as follows :

$$\widetilde{\nabla} u_i = \frac{\sum_{\substack{T|i \in T \\ T \in fluid}} \nabla u_h |T|}{\sum_{\substack{T|i \in T \\ T \in fluid}} |T|} \quad (2.111)$$

This modification is only present to be consistent for the variable derivatives and thus do not impact that much the solution looking at primitive variables. However, as soon as derivatives intervene, results are affected considerably, as we will see section 3.2.5.2.

So as to prove the robustness of the proposed extension of the RD framework to the Penalization, several examples are proposed in the next section.

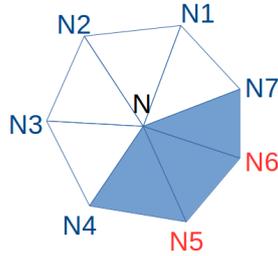


Figure 2.12 – Boundary point needing special treatment for the gradient reconstruction.

### 2.3.5 Penalized Navier Stokes Results

In this section, numerical examples are employed to validate the proposed resolution of the penalized NS equations. First, two  $2D$  test cases are presented. The first one is a classic study of the flow around a naca0012 airfoil. The second one is the study of a supersonic flow around a wedge. It will emphasize the possibility to handle complex flow with shock. The result will be validated by comparison with literature data. A  $3D$  test case is also presented (without validation), to prove the feasibility of the method in  $3D$ . It corresponds to the extrusion of the  $2D$  wedge into the  $z$  direction.

#### 2.3.5.1 2D Naca0012 airfoil

The naca0012 airfoil of characteristic length 1 is placed in a circular domain of radius  $r = 20$ . The simulation is performed for a Reynolds number  $Re = 5000$ . Only the velocities are penalized to 0. The inflow/outflow parameters are given by the following set of dimensionless variables :  $\rho = 1$ ,  $u = 0.5$ ,  $v = 0$  and  $p = 1/\gamma$ .

As explained section 2.2.4, the initial mesh is adapted to the 0 isovalue of the level set (see figure 2.13a). To adapt the mesh, the physical variable chosen is the  $u$  velocity. For the adaptation, the parameters to compute the metrics (2.40) and (2.42) are presented table 2.1. A total of 4 iterations of the iteration procedure presented section 2.2.4 are performed. The resulted mesh with isolines of  $u$  velocity are plotted figure 2.13d and compared to the solution on initial mesh figure 2.13c. A zoom on the mesh close to the solid is proposed figure 2.13b.

As expected, thanks to the adaptation, we considerably increase the accuracy of the solution. Figures 2.13b and 2.13d show that the wake is well captured (till the boundary of the domain), and that the level set adaptation in addition of performing a good description of the solid boundary proposes a suitable mesh for the resolution of the boundary layer.

Level set adaptation	Physical adaptation
$\epsilon = 0.001$	$\epsilon = 0.001$
$h_{min} = 0.0005$	$h_{min} = 0.001$
$h_{max} = 2.0$	$h_{max} = 2.0$
$w = 0.0005$	

Table 2.1 – Adaptation parameter for Naca0012 airfoil

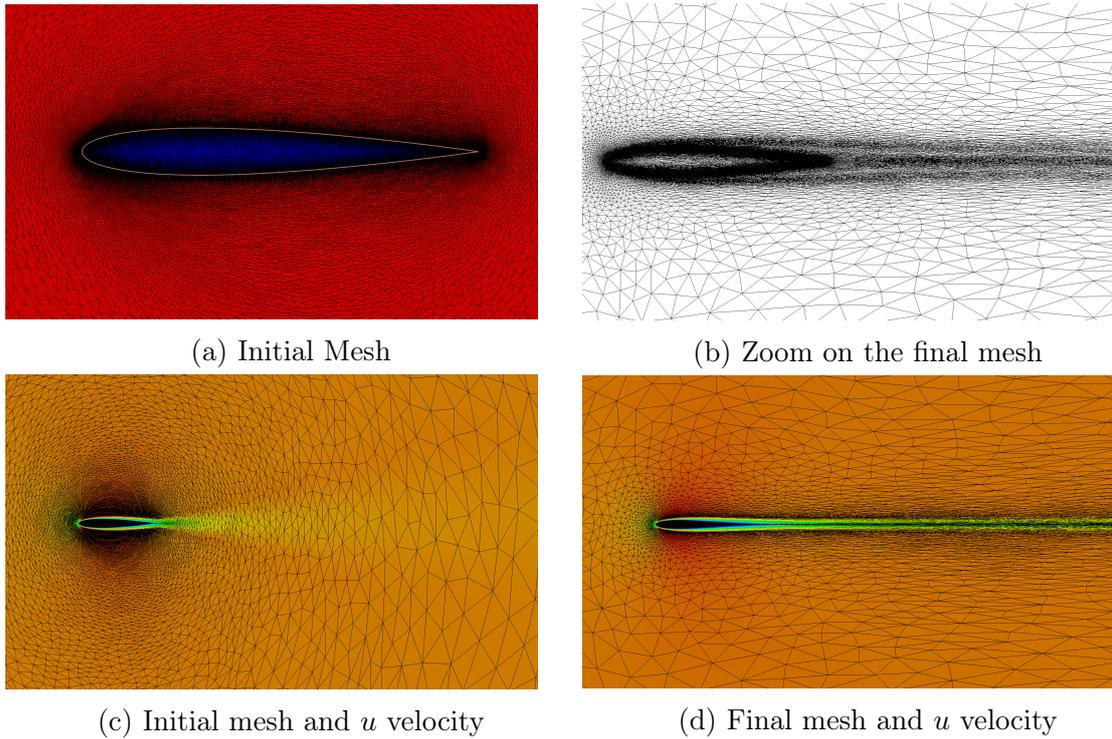


Figure 2.13 – Naca0012 test case

A comparison is performed against a body fitted simulation, the mesh being adapted with the same criteria. Figures 2.14 proposes the comparison between the mesh and solution in a vicinity of the airfoil. To compare more precisely the solution at the wall, we perform two cuts, at  $x = 0.25$  and  $x = 0.5$  and we plot the density,  $u$  velocity and temperature figures 2.15a, 2.15b and 2.15c. Those results show that the imposition of the BC with the penalization compares well w.r.t. the imposition of the BC on fitted mesh for "classical" simulation. Thus, the penalization presents a competitive alternative to classical fitted simulations.

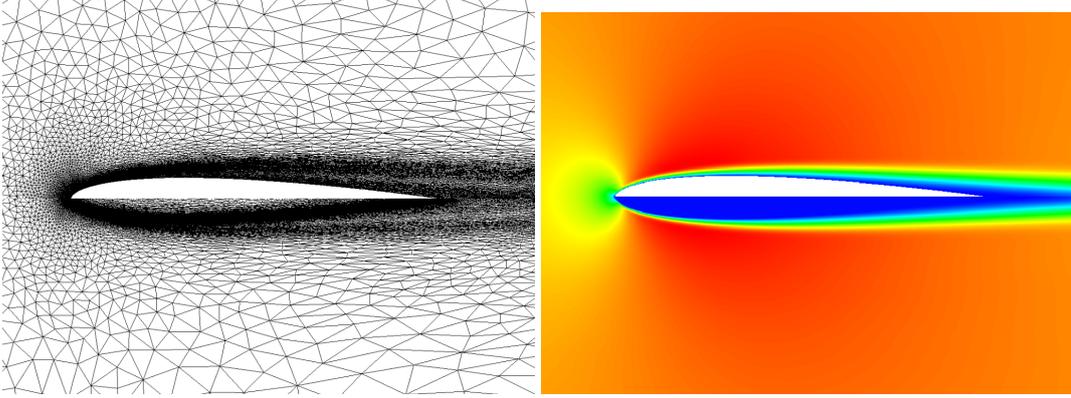


Figure 2.14 – Naca0012 : Comparison fitted Penalization. Left : Meshes - Right :  $u$  velocity

Level set adaptation	Physical adaptation
$\epsilon = 0.0005$	$\epsilon = 0.001$
$h_{min} = 0.0005$	$h_{min} = 0.0005$
$h_{max} = 2.0$	$h_{max} = 2.0$
$w = 0.0005$	

Table 2.2 – Adaptation parameter for 2D wedge

### 2.3.5.2 2D supersonic flow past wedge

We consider the supersonic flow around a wedge of height  $h = 0.5$ , half angle  $\theta = 20$  deg (see figure 2.16a). The wedge is placed in a circle of radius 20. We show the results obtained with the Lax-Friedrich method. The flow is characterized by a Reynolds number  $Re = 50000$ , a Mach number  $Ma = 2$  (chosen such that the shock is in contact with the wedge), and a non-dimensional pressure 1. The velocities are penalized to  $u = v = 0$  and the temperature to  $T_S = 3$ . For the adaptations, the parameters used are regrouped table 2.2. The initial mesh (30407 vertices) is presented figure 2.17a, and the one obtained after 5 cycles of adaptations (111061 vertices) w.r.t. the  $u$ -velocity is presented figure 2.17b. A zoom near the interface is presented for both initial and final mesh figures 2.17c and 2.17d.

To validate our results, we compare the computed shock angle with its analytical value. As in [32], the angle is measured using a point located on the shock near  $y = 0$  and we found  $\beta = 53.33$  deg for a analytic one of  $\beta \approx 53.46$  deg. Our results are in good agreement with the theory and the numerical solutions performed in [32].

## 2. Steady Penalized Navier Stokes Equations

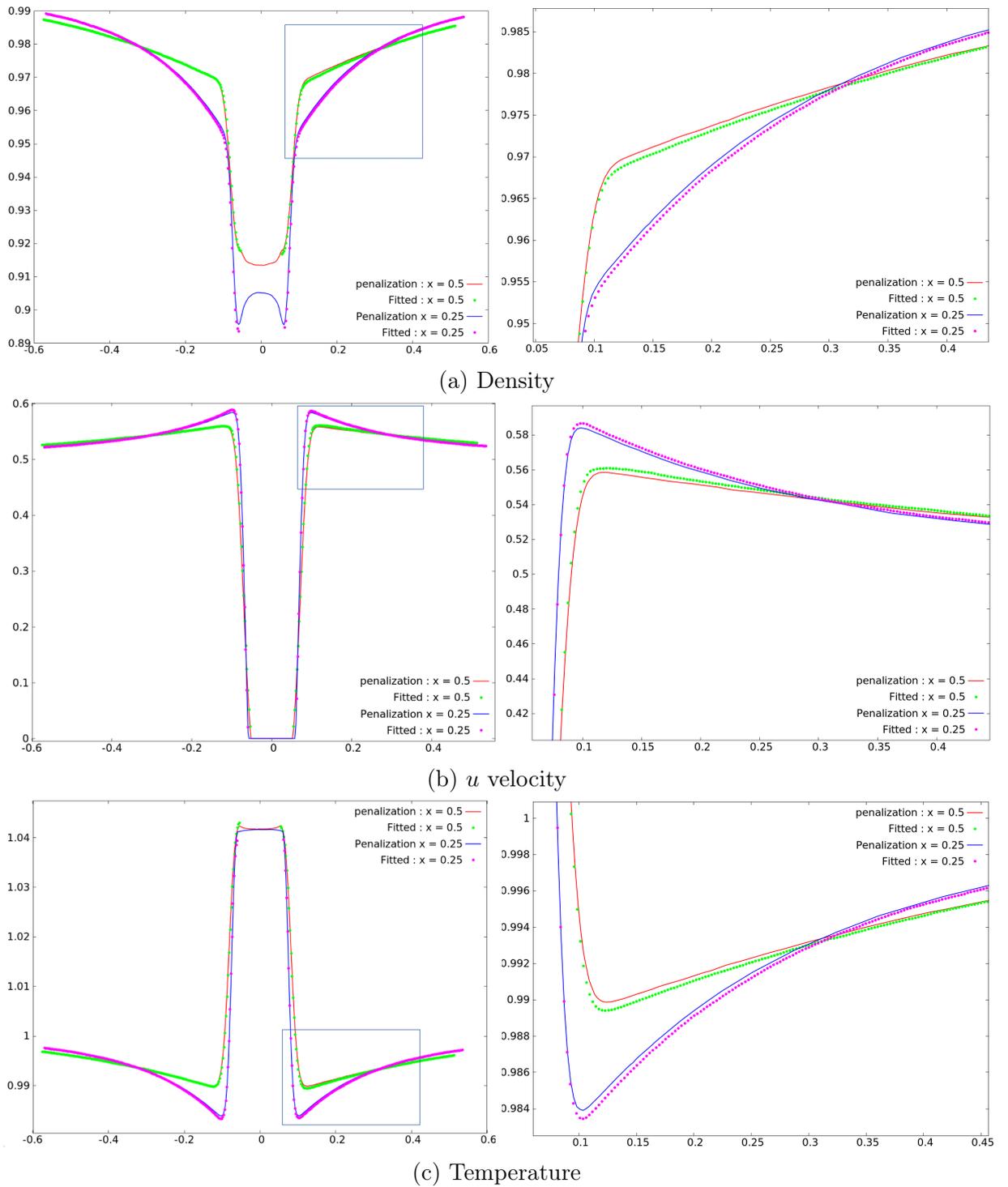


Figure 2.15 – Plot at  $x = 0.25$  and  $x = 0.5$ . Zoom on the right

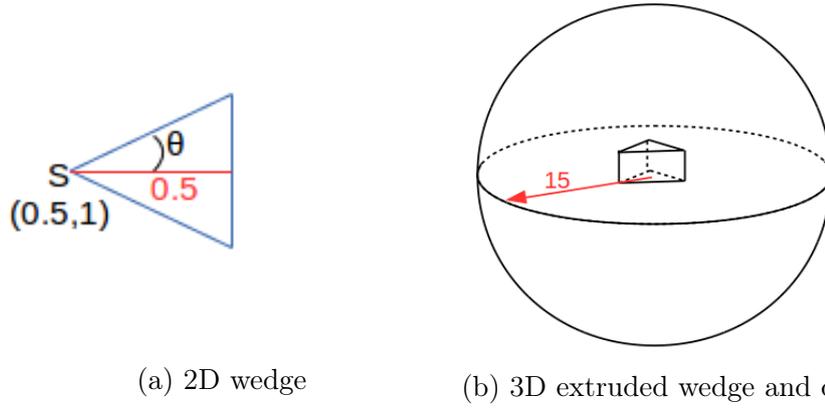


Figure 2.16 – 2D and 3D wedge test cases

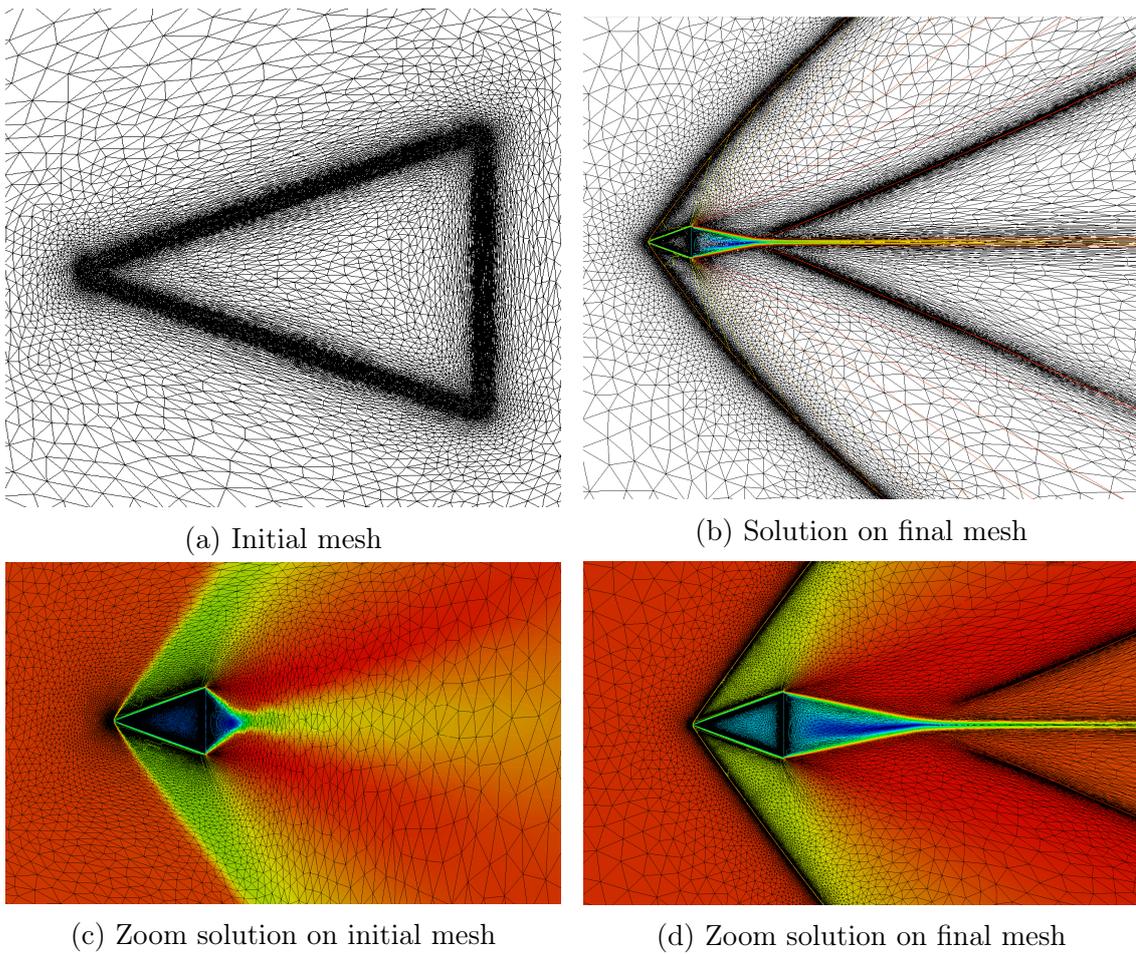


Figure 2.17 – 2D wedge test case results

Level set adaptation	Physical adaptation
$\epsilon = 0.0005$	$\epsilon = 0.0005$
$h_{min} = 0.005$	$h_{min} = 0.005$
$h_{max} = 2.0$	$h_{max} = 2.0$
$w = 0.0005$	

Table 2.3 – Adaptation parameter for 3D wedge

### 2.3.5.3 3D supersonic flow past extruded wedge

This 3D test case is the extension in 3D of the previous one. The wedge is extruded according to the  $z$  direction of a size 0.364 and the domain is a sphere of radius 15 (see figure 2.16b). All the data for the computation are the same than in 2D plus the  $w$  velocity (component according to the  $z$  direction) which is 0 on the BC. As previously, the initial mesh (36,597 vertices and 216,703 tetrahedra) is only adapted to the 0 level set (see figure 2.18c for a cut in the  $(xy)$  plan described figure 2.18b). The mesh obtained after 5 cycles of adaptation (766310 vertices and 4,545,447 tetrahedra) is presented figure 2.18d. To generate those meshes, the parameters employed are regrouped table 2.3.

On figures 2.18e and 2.18f,  $u$  velocity solution on the initial and adapted mesh are compared at a cut in the  $(xz)$  plan (figure 2.18a). As in 2D, we considerably improve the solution, as well on the shock definition than in the drag which is really better defined. And this, by limiting the number of inserted nodes.

## 2.4 Summary

In this chapter, the resolution of the steady penalized Navier Stokes equations has been addressed. The different test cases proposed aimed at demonstrate the competitive alternative that represents the penalization on adaptive grids with respect to classical fitted resolution of the Navier Stokes equations. To solve the equations, the residual distribution framework has been extended to the penalization by the use of a simple Galerkin discretization of the penalty source term. Metric based mesh adaptation has been employed to adapt the meshes, leading to an increase of the accuracy to the solid boundary definition and the physical solution. To exploit the full potential of the penalization, unsteady cases can now be studied. However, before considering moving objects, the background for unsteady problems is addressed in the next chapter.

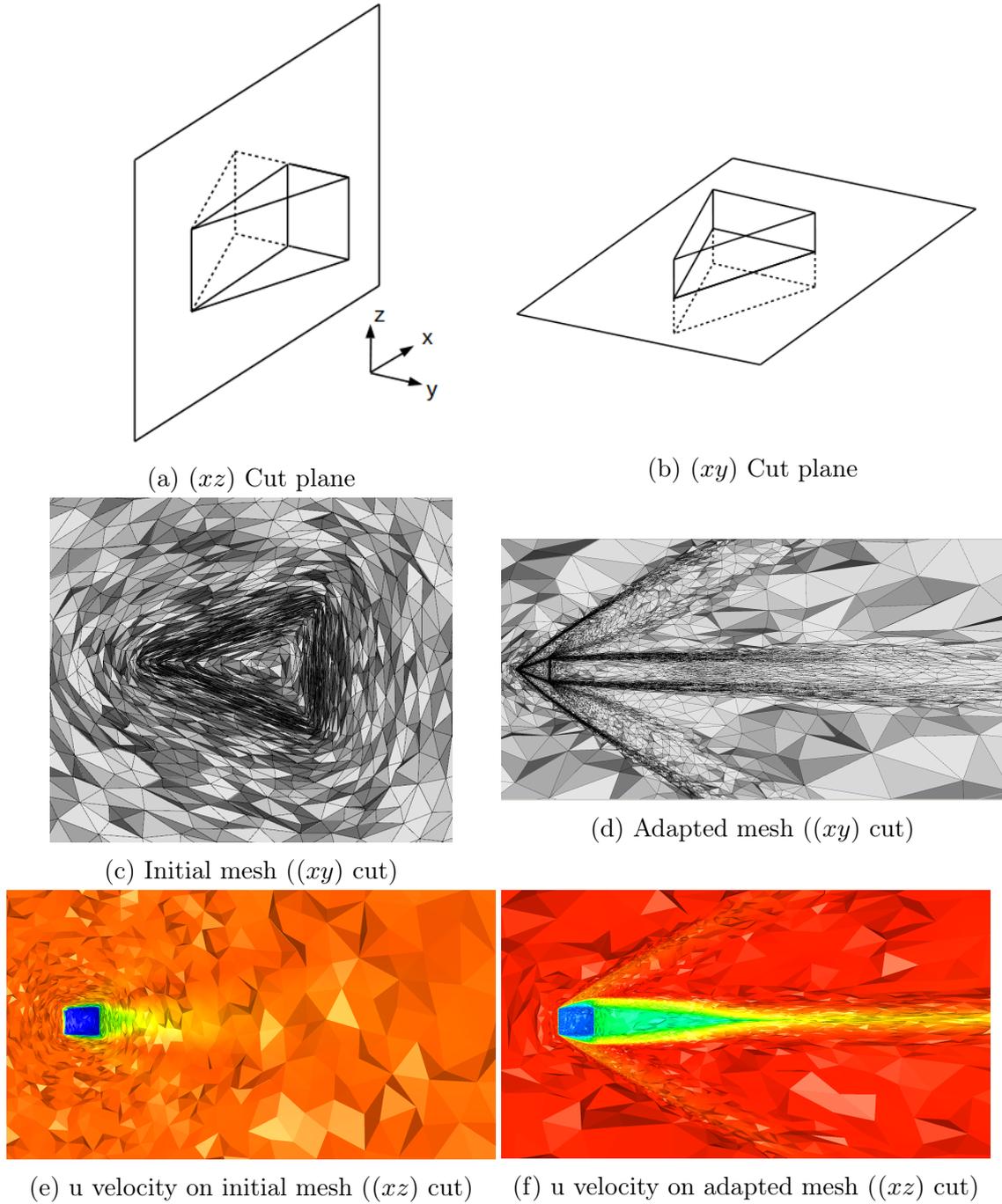


Figure 2.18 – 3D wedge test case results

## Chapter 3

# Unsteady Penalized Navier Stokes Equations For Motionless Bodies

The focus of this chapter is the resolution of unsteady problems, but for motionless bodies, so as to provide numerical tools to deal with unsteady penalized simulations. The schemes introduced previously have never been employed for unsteady problems, even without penalization, or for simpler problem as *e.g.* the time dependant advection diffusion equation. So, we dedicate a first section to the extension and validation of the schemes to these unsteady cases. Then, we propose a strategy to combine these schemes to penalization. Indeed, the process differs from steady problems for which an implicit Euler time discretization is employed to solve the pseudo unsteady iterations. In the time dependant case, a splitting approach is proposed, leading to efficient schemes and to results proposed at the end of this chapter.

## 3.1 Residual Schemes for Unsteady Problems

### 3.1.1 State of the art

The residual distribution framework has been extended to time dependant advection problems by means of two different approaches. Space time schemes have been proposed in Csík [51] and Mezine's PhD [116] and later in Ricchiuto's PhD [133] (see also [52, 139]) and Villedieu's PhD [160]. More recent work following this approach can be found in [89] where unconditionally positive schemes are proposed and linearized Euler equations are solved in [105]. The time contribution has also been taken into account *via* mass matrices, idea introduced in [112] and [69] at the Von Karman Institute, and to perform LES simulations in Caranei's PhD [41]. This approach has also been studied by Mezine and Abgrall [13, 116] and De Palma *et. al.* [128]. More recently, genuinely explicit schemes have been proposed in [135]. All of those works mentioned application to the Euler equations when considering system case. Another application, the Shallow Water system (even if related to Euler problem), has been considered by Hubbard and co authors [120, 87], and more recently in [138, 137]. A global overview of those work can be found in the *HDR* of Ricchiuto [134], that is the starting point of the present work.

The study of time dependent advection diffusion equations is less documented. As said before, Caranei employed RD schemes to perform LES simulations ([41, 39, 40]). Interesting results are shown by Dobes *et. al.*, who extended the work already mentioned section 2.3.1 for steady simulations ([136]) using an hybrid RD-Galerkin scheme considering local Peclet number [61]. Based on the FOS approach, Mazaheri and Nishikawa obtained very recently schemes up to 6 order accurate validated on 1D problems in [114].

In this work, the theory developed for advection problem is combined to the theory proposed in De Santis' PhD [56] recalled section 2.3.3.5. As will be seen in the next sections, the spatial fluctuation is computed using the reconstructed gradient plus the penalized term coming from the FOS and second order schemes are constructed.

### 3.1.2 Principle

As for steady case, we start by explaining the construction and recalling properties for a 2D scalar advection equation (2.30). As in section 2.3.3, the Jacobian of the advective flux is denoted  $\mathbf{a} = \nabla_u \mathcal{F}$ , and a mesh  $\mathcal{T}^h$ , discretization of  $\Omega$  is considered. The time interval is also decomposed as  $[0, T_f] = \cup_{i=0, N-1} [t^i, t^{i+1}]$  with  $t^0 = 0$  and  $t^N = T_f$ . As presented section 2.3.2, the approximated solution is defined using the  $P^1$  Lagrange basis function  $u_h(t) = \sum_i u_i(t) \varphi_i$  and by denoting  $u^n = u(t^n)$  we define  $u_h^n = \sum_i u_i^n \varphi_i$ .

A  $(r + 1)^{th}$  order time discretization of this equation (2.30) writes over the general form :

$$\Gamma^{n+1}(u) = \sum_{i=0}^p \alpha_i \frac{\delta u^{n+1-i}}{\Delta t} + \sum_{j=0}^q \theta_j \nabla \cdot \mathcal{F}^{n+1-j} \quad (3.1)$$

with  $\delta u^{n+1} = u^{n+1} - u^n$ ,  $\mathcal{F}^{n+1-j} = \mathcal{F}^{n+1-j}(u^{n+1-j})$ ,  $\alpha_i$  and  $\theta_j$  coefficients dependant of the chosen scheme.

The *total residual* is then defined on each element by :

$$\Phi^T(u_h) = \int_T \Gamma^{n+1}(u_h) = \int_T \left( \sum_{i=0}^p \alpha_i \frac{\delta u^{n+1-i}}{\Delta t} + \sum_{j=0}^q \theta_j \nabla \cdot \mathcal{F}^{n+1-j} \right) \quad (3.2)$$

This total residual can be linked to the fluctuation  $\phi^T(u_h)$  (2.46) defined for steady problem by :

$$\Phi^T(u_h) = \int_T \left( \sum_{i=0}^p \alpha_i \frac{\delta u^{n+1-i}}{\Delta t} \right) + \sum_{j=0}^q \theta_j \phi^T(u_h^{n+1-j})$$

Similarly, for the boundaries  $\partial T$  such that  $\partial T \subset \partial \Omega$ , it is set :

$$\phi^{\partial T} = \int_{\partial T} \sum_{j=0}^q \theta_j (\mathcal{F}^{BC} - \mathcal{F}_h)^{n+1-j} \cdot \mathbf{n} \quad (3.3)$$

with  $\mathcal{F}^{BC}$  a numerical flux consistent with the chosen BC.

Then, the same principle than for steady case is applied. The aim is to define *nodal residual*  $\Phi_i^T(u_h)$  satisfying :

$$\sum_{i \in T} \Phi_i^T(u_h) = \Phi^T(u_h), \quad \sum_{i \in \partial T} \phi_i^{\partial T} = \phi^{\partial T} \quad (3.4)$$

and the RD scheme writes :

$$\sum_{T \ni i} \Phi_i^T(u_h) + \sum_{\partial T \ni i} \phi_i^{\partial T} = 0 \quad (3.5)$$

### 3.1.3 Basic Properties

In this section, usual properties proposed in [134] are recalled. It starts by an assumption concerning the mesh and the time stepping strategy :

**Property 3.1.**

$$C_0 \leq \sup_{K \in \mathcal{T}^h} \frac{h^2}{|K|} \leq C_1, \quad C'_0 \leq \frac{\Delta t}{h} \leq C'_1 \quad (3.6)$$

with  $\Delta t = \min_n (t^{n+1} - t^n)$ .

The accuracy of a RD scheme for time dependant case is then defined as follows :

**Definition 3.1. Accuracy of RD scheme, time dependent case**

Under assumption (3.1), let consider a  $(r+1)^{th}$  order time integration scheme, and  $(k+1)^{th}$  order accurate approximation of the flux and unknowns.  $\psi \in C_0^1(\Omega \times [0, T_N])$  is a smooth test function with  $\psi|_{\partial\Omega} = 0$  and  $\psi_h$  its  $(k+1)^{th}$  accurate approximation. Defining the following truncation error :

$$\epsilon(u_h, \psi) := \sum_{n=0}^N \sum_{i \in \mathcal{T}^h} \Delta t^{n+1} \psi_i^{n+1} \sum_{T \ni i} \Phi_i^T(u_h) = \sum_{n=0}^N \sum_{K \in \mathcal{T}^h} \sum_{i \in T} \int_{t^n}^{t^{n+1}} \psi_i^{n+1} \Phi_i^T(u_h)$$

a RD scheme is said  $(p+1)^{th}$  order accurate if

$$|\epsilon(u_h, \psi)| \leq \mathcal{O}(h^{p+1}), \quad p = \min(k, r) \quad (3.7)$$

And the following property holds, whom demonstration can be found in [134] :

**Property 3.2. Accuracy Condition**

A RD scheme defined as (3.5) is  $(p+1)^{th}$  order accurate in the sense of definition 3.1 provided that its resulting approximated solution  $u_h$  satisfies :

$$\sup_{T \in \mathcal{T}^h} \sup_{i \in T} |\Phi_i^T(u_h)| = \mathcal{O}(h^{p+2}) \quad (3.8)$$

As proposed section 2.3.3.3 for steady case, RD schemes can be seen as a reformulation of stabilized FE formulation, resumed by the following property :

**Property 3.3. High order residual schemes**

Under the hypothesis of property 3.2, a sufficient condition for a scheme defined as (3.5) to be  $(p+1)^{th}$  order accurate is, if there exists a function  $\omega_i$  uniformly bounded w.r.t.  $h$ ,  $u_h$ ,  $\Gamma^{n+1}(u_h)$  and the data of the problem such that :

$$\Phi_i^T(u_h) = \int_T \omega_i \Gamma^{n+1}(u_h) \quad (3.9)$$

From now on, when expressing the RD scheme over a stabilized finite element formulation, the operators  $\mathcal{L}_h^G$  and  $\mathcal{L}_h^S$  will be used, denoting respectively the Galerkin part of the residual and the stabilised one :

$$\begin{cases} \mathcal{L}_h^G(u_h) + \mathcal{L}_h^S(u_h) = 0 \\ \mathcal{L}_h^G(u_h) = \sum_{T \ni i} \int_T \varphi_i \left( \sum_{i=0}^p \alpha_i \frac{\delta u^{n+1-i}}{\Delta t} + \sum_{j=0}^q \theta_j \nabla \cdot \mathcal{F}^{n+1-j} \right) \\ \mathcal{L}_h^S(u_h) = \sum_{T \ni i} \int_T \gamma_i \left( \sum_{i=0}^p \alpha_i \frac{\delta u^{n+1-i}}{\Delta t} + \sum_{j=0}^q \theta_j \nabla \cdot \mathcal{F}^{n+1-j} \right) \end{cases} \quad (3.10)$$

### 3.1.4 Mass Matrix Formulations

Let consider the simplest case of (3.1), a one step time integration scheme, with  $p = 0$  and  $\alpha_0 = 1$ . For sake of simplicity in the notations, the spatial contribution is denoted with  $\phi_i^T(u_h)$ . Then, the nodal residual writes :

$$\Phi_i^T(u_h) = \int_T \omega_i \frac{\Delta u_h}{\Delta t} + \phi_i^T(u_h) \quad (3.11)$$

where  $\Delta u_h = u_h^{n+1} - u_h^n$ . This can be rewritten as :

$$\Phi_i^T(u_h) = \sum_{j \in T} \frac{\Delta u_j}{\Delta t} \int_T \omega_i \varphi_j + \phi_i^T(u_h) \quad (3.12)$$

and thus the mass matrix ( $m_{ij}$ ) is introduced :

$$\Phi_i^T(u_h) = \sum_{j \in T} m_{ij} \frac{\Delta u_j}{\Delta t} + \phi_i^T(u_h), \quad m_{ij} = \int_T \omega_i \varphi_j \quad (3.13)$$

As detailed in [134, 135], different formulations can be sorted out for this mass matrix. In this work, two different definitions are employed. The first one is defined by using the distribution coefficient defined for steady case, but to distribute not only fluctuation (2.46), but the total residual (3.2). It defines the nodal residual as :

$$\Phi_i^T = \beta_i^T \left( \sum_{j \in T} \int_T \frac{\Delta u_h}{\Delta t} + \phi^T(u_h) \right) \quad (3.14)$$

leading to the formulation 1 of the mass matrix :

$$m_{ij}^{F1} = \beta_i^T \frac{|T|}{d+1}, \quad \omega_i = \beta_i^T \quad (3.15)$$

The second formulation used is to consider  $\omega_i$  as defined section 2.3.3.3 for steady case (2.71,2.73) :

$$\omega_i = \varphi_i + \beta_i^T - \frac{1}{N_{DoF}} \quad (3.16)$$

Using this definition (3.16) in (3.13) gives the formulation 2 of the mass matrix :

$$m_{ij}^{F2} = \frac{|T|}{d+1} \left( \frac{1 + \delta_{ij}}{d+2} + \beta_i^T - \frac{1}{d+1} \right) \quad (3.17)$$

**Remark 3.1.** As the  $\beta_i^T$  are defined satisfying the Linearity Preservation property 2.6 those two definitions of  $\omega_i$  satisfy the property 3.3.

Now that all the theory is settled, different discretizations are now presented, and applied to advection diffusion equations, by combining this theory developed and already validated for advection problems with the fluctuation defined previously for steady advection diffusion problems.

### 3.1.5 Discretization of scalar advection diffusion problem

The focus is now put on the discretization of time dependant advection diffusion equation (2.33): As said in the state of the art, some works have been provided in the literature [39, 61]. The present approach extends the schemes based on the gradient reconstruction strategy proposed in [56, 11] recalled section 2.3.3.5 to unsteady simulations. The schemes are written exactly as for time dependant advection problems and the total and nodal residuals are defined as for steady case when switching to advection diffusion problems. First, the second order explicit Runge Kutta scheme proposed by Ricchiuto and Abgrall in [135] is briefly recalled and extended to the considered problems. Indeed, as will be explained, they provide a scheme that can be fully explicit, without even the inversion of a Galerkin mass matrix. Thus, its implementation is very simple and even if, as every explicit scheme, the time step is constrained by a CFL condition, it still allows to perform sufficient quick simulations to first settle the residual distribution framework applied to penalized Navier Stokes equations. Nevertheless, when coming to perform real complex simulations on adapted grids, the use of an implicit scheme becomes necessary. Thus, the  $\theta$  scheme is also presented, that is used with  $\theta = 0.5$ , giving a Crank-Nicolson (CN) scheme, second order accurate in time.

#### 3.1.5.1 Explicit Second Order Runge Kutta Scheme

As explained in the introduction, genuinely explicit schemes have been proposed by Ricchiuto and Abgrall [135, 134] and used for solving advection problems. Here is briefly recalled the construction of the RK2 scheme, the reader being referred to the above references for proof of accuracy.

Following [135], the construction starts from the stabilized FE formulation :

$$\mathcal{L}_h^G(u_h) + \mathcal{L}_h^S(u_h) = \sum_{T \ni i} \left\{ \int_T \varphi_i \frac{\Delta u_h}{\Delta t} + \int_T \gamma_i \frac{\Delta u_h}{\Delta t} + \int_T \omega_i \nabla \cdot \mathcal{F}(u_h) \right\} = 0 \quad (3.18)$$

What has been proved in the above references is that what is called a *time shifted operator* (TSO)  $\overline{\Delta u_h}$  can be used in the stabilized part :

$$\mathcal{L}_h^S(u_h) = \sum_{T \ni i} \int_T \gamma_i \left( \frac{\overline{\Delta u_h}}{\Delta t} + \nabla \cdot \mathcal{F}(u_h) \right) \quad (3.19)$$

This TSO being chosen genuinely simplifies a lot the problem without damaging the accuracy of the scheme. It allows to only have to invert the Galerkin matrix (or, as will be seen, avoid matrix inversion using mass lumping). For the RK2 scheme, the two following TSO have been proven to conserve the

second order accuracy :

$$\begin{cases} \overline{\Delta u_h}^{rk1} = 0 \\ \overline{\Delta u_h}^{rk2} = \frac{u_h^* - u_h^n}{\Delta t} \end{cases} \quad (3.20)$$

where  $u_h^*$  denotes the approximation of the solution after the first step of the scheme.

Thus, the first step writes :

$$\sum_{T \ni i} \left\{ \int_T \varphi_i \frac{\Delta u_h}{\Delta t} + \int_T \omega_i \nabla \cdot \mathcal{F}(u_h) \right\} = 0 \quad (3.21)$$

leading to:

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij}^G \frac{u_j^* - u_j^n}{\Delta t} + \phi_i^T(u_h^n) \right\} = 0 \quad (3.22)$$

where  $m_{ij}^G = \int_T \varphi_i \varphi_j$  corresponds with the galerkin matrix and  $\phi_i^T(u_h)$  corresponds with the nodal fluctuation (2.74). Even if the Galerkin mass matrix is easy to invert, so as to use full potential of explicit scheme, and as the focus of this PhD is second order accuracy, the choice has been done to use mass lumping, leading to the following first step of the RK2 scheme :

$$\frac{|C_i|}{\Delta t} (u_i^* - u_i^n) + \sum_{T \ni i} \phi_i^T(u_h^n) = 0. \quad (3.23)$$

Identically, the second step is written :

$$\sum_{T \ni i} \left\{ \int_T \varphi_i \frac{u_h^{n+1} - u_h^n}{\Delta t} + \int_T \gamma_i \frac{u_h^* - u_h^n}{\Delta t} + \frac{1}{2} \int_T \omega_i [\nabla \cdot \mathcal{F}(u_h^n) + \nabla \cdot \mathcal{F}(u_h^*)] \right\} = 0 \quad (3.24)$$

Once again, as only second order accuracy is sought, for sake of simplicity in the implementation, it is used :

$$\frac{1}{2} (\mathcal{F}(u_h^*) + \mathcal{F}(u_h^n)) = \mathcal{F} \left( \frac{u_h^* + u_h^n}{2} \right)$$

To get a simpler form of (3.24), the Galerkin integral  $\int_T \varphi_i \frac{u_h^* - u_h^n}{\Delta t}$  is added and subtracted :

$$\sum_{T \ni i} \left\{ \int_T \varphi_i \frac{u_h^{n+1} - u_h^n}{\Delta t} + \int_T \gamma_i \frac{u_h^* - u_h^n}{\Delta t} + \int_T \omega_i \nabla \cdot \mathcal{F} \left( \frac{u_h^n + u_h^*}{2} \right) \pm \int_T \varphi_i \frac{u_h^* - u_h^n}{\Delta t} \right\} = 0$$

leading to :

$$\sum_{T \ni i} \left\{ \int_T \varphi_i \frac{u_h^{n+1} - u_h^*}{\Delta t} + \int_T \omega_i \frac{u_h^* - u_h^n}{\Delta t} + \int_T \omega_i \nabla \cdot \mathcal{F} \left( \frac{u_h^n + u_h^*}{2} \right) \right\} = 0$$

that written in a RD formalism gives :

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij}^G \frac{u_h^{n+1} - u_h^*}{\Delta t} + \sum_{j \in T} m_{ij} \frac{u_h^* - u_h^n}{\Delta t} + \phi_i^T \left( \frac{u_h^* + u_h^n}{2} \right) \right\} = 0$$

As for the first step, mass lumping is applied and finally, the RK2 RD scheme writes, with the formulation 1 of the mass matrix :

$$\begin{cases} \frac{|C_i|}{\Delta t} (u_i^* - u_i^n) + \sum_{T \ni i} \phi_i^T(u_h^n) = 0 \\ \frac{|C_i|}{\Delta t} (u_i^{n+1} - u_i^*) + \sum_{T \ni i} \left[ \sum_{j \in T} m_{ij}^{F_1} \frac{u_j^* - u_j^n}{\Delta t} + \phi_i^T \left( \frac{u_h^* + u_h^n}{2} \right) \right] = 0 \end{cases} \quad (3.25)$$

The extension to advection diffusion problems is quite simple. The scheme (3.25) is modified by defining the nodal fluctuation as for steady case. The fluctuation is computed using the reconstructed gradient  $\widetilde{\nabla} u_h$  and the penalty term coming from the FOS is added to the nodal fluctuation. Considering as previously the simplest case for the diffusion part,  $\mathbf{K} = \nu \mathbb{I}$  :

$$\begin{aligned} \phi_i^T(u_h) &= \beta_i^T \int_T \left( \nabla \cdot \mathcal{F}(u_h) - \nu \nabla \cdot \widetilde{\nabla} u_h \right) + \int_T \nu \nabla \varphi_i \cdot \left( \nabla u_h - \widetilde{\nabla} u_h \right) \\ &= \beta_i^T \phi^T(u_h) + \int_T \nu \nabla \varphi_i \cdot \left( \nabla u_h - \widetilde{\nabla} u_h \right) \end{aligned} \quad (3.26)$$

with  $u_h = u_h^n$  at the first step and  $u_h = \frac{u_h^n + u_h^*}{2}$  at the second one.

### 3.1.5.2 Implicit $\theta$ scheme

#### A Newton Algorithm

Still considering the unsteady advection diffusion equation (2.33), let consider the  $\theta$  scheme written in the RD formalism (3.13) :

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij} \frac{u_j^{n+1} - u_j^n}{\Delta t} + \theta \phi_i^T(u_h^{n+1}) + (1 - \theta) \phi_i^T(u_h^n) \right\} = 0, \quad \forall i \in \mathcal{T}^h \quad (3.27)$$

As for steady simulation, the non linear character of the problem and the need of the evaluation at time  $t^{n+1}$  of the fluctuation  $\phi_i^T(u_h^{n+1})$  requires the use of Newton algorithm. As for the RK2 scheme, considering second order accuracy (at best,  $\theta = \frac{1}{2}$ , Crank-Nicolson (CN RD) scheme), it is used :

$$\phi_i^T(u_h^{n+\theta}) = \theta \phi_i^T(u_h^{n+1}) + (1 - \theta) \phi_i^T(u_h^n), \quad u_h^{n+\theta} = \theta u_h^{n+1} + (1 - \theta) u_h^n \quad (3.28)$$

Let rewrite (3.27) as :

$$\begin{cases} R_i(u_h^{n+1}) = 0, \quad \forall i \in \mathcal{T}^h \\ R_i(u_h) = \sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij} \frac{u_j - u_j^n}{\Delta t} + \phi_i^T(u_h^{n+\theta}) \right\} \end{cases} \quad (3.29)$$

The iterative convergence of

$$\begin{cases} R_i(u_h^{k+1}) = 0 \\ u_h^0 = u_h^n \end{cases} \quad (3.30)$$

is sought. The following Taylor expansion is employed :

$$R_i(u_h^{k+1}) \approx R_i(u_h^k) + \sum_{j \in \mathcal{T}^h} \frac{\partial R_i}{\partial u_j^k} (u_j^{k+1} - u_j^k)$$

And the problem writes :

$$\begin{cases} \underline{\mathbf{J}}(\mathbf{U}^k) (\mathbf{U}^{k+1} - \mathbf{U}^k) = -\mathbf{R}(\mathbf{U}^k) \\ \mathbf{U}^0 = \mathbf{U}_h^n \end{cases} \quad (3.31)$$

where  $\mathbf{U} = (u_0, \dots, u_{N_v})^T$ ,  $\mathbf{R}(\mathbf{U}) = (R_0(u_h), \dots, R_{N_v}(u_h))^T$  and with  $\underline{\mathbf{J}}(\mathbf{U}^k)$  the Jacobian to invert.

$$J_{ij} = \begin{cases} \sum_{T \ni i} \frac{\partial R_i}{\partial u_i^k}, & \text{if } i = j \\ \sum_{T \ni (i,j)} \frac{\partial R_i}{\partial u_j^k}, & \text{if } i \neq j \end{cases} \quad (3.32)$$

### Jacobian approximation

Now, the question raised is the evaluation of this Jacobian. Indeed, some terms are neglected so as to simplify the expression and the computation of the matrix to invert, especially for the diffusive part.

$R_i$  is rewritten as :

$$R_i(u_h) = \sum_{T \ni i} \left\{ \underbrace{\sum_{j \in T} m_{ij} \frac{u_j - u_j^n}{\Delta t}}_{R_{i_1}} + \underbrace{\phi_i^T(u_h^{n+\theta})}_{R_{i_2}} \right\} \quad (3.33)$$

Starting by considering  $R_{i_2}$ , the nodal fluctuation is splitted into the advective contribution  $\phi_i^{adv}$  and the diffusive one  $\phi_i^{diff}$ ,  $\phi_i^T = \phi_i^{adv} + \phi_i^{diff}$  with :

$$\phi_i^{adv} = \beta_i^T \phi^{T,adv}(u_h) = \beta_i^T \sum_{j \in T} k_j u_j \quad (3.34)$$

$$\phi_i^{diff} = -\beta_i^T \int_T \nabla \cdot (\nu \nabla u_h) + \int_T \nu \nabla \varphi_i \cdot (\nabla u_h - \widetilde{\nabla} u_h) \quad (3.35)$$

**Remark 3.2.** The advective part is defined in the Jacobian using the inflow parameters, coming from the quasi linear form of the equation (2.55), as the aim is to find an approximation. For the right and side, however, the computation is still performed via the integration of the flux on the boundary (2.54) so as to keep a conservative form.

First, looking at the advective nodal residual (3.34), the first approximation consists in neglecting the Jacobians of the distribution coefficients  $\beta_i^T$  and inflow parameters  $k_i$ . Thus, the Jacobian of the advective part simply writes :

$$\frac{\partial \phi_i^{T,adv}}{\partial u_j} = \beta_i^T k_j \quad (3.36)$$

The second approximation concerns the diffusive part of the fluctuation. Only the advection Jacobian could be used to solve the whole problem. However, so as to ensure a better convergence, it is chosen here to take the diffusion into account in the matrix. Nevertheless, regarding (3.35), the diffusion contribution of the fluctuation being computed using reconstructed gradients, its Jacobian is complex to evaluate. An idea could be to consider a  $P1$  approximation of the gradient, but it would lead to a null contribution. Thus, the choice has been done to only and simply consider the classical Galerkin matrix associated to diffusion :  $\sum_{j \in T} u_j \nu \int_T \nabla \varphi_i \cdot \nabla \varphi_j$  leading to the final approximation of the Jacobian of  $R_{i_2}$  :

$$\frac{\partial R_{i_2}}{\partial u_j} = \beta_i^T k_j + \nu \int_T \nabla \varphi_i \cdot \nabla \varphi_j = \beta_i^T k_j + \nu \frac{\mathbf{n}_i \cdot \mathbf{n}_j}{d^2 |T|} \quad (3.37)$$

Now, regarding  $R_{i_1}$ , as the mass matrix is non linear only by its dependence on the distribution coefficients (whatever is the chosen formulation, see definitions (3.15,3.17)), its Jacobian is also neglected. Thus, the Jacobian of  $R_{i_1}$  writes :

$$\frac{\partial R_{i_1}}{\partial u_j} = \frac{m_{ij}}{\Delta t} \quad (3.38)$$

Combining those two approximations 3.38 and 3.37, the final form of the matrix to invert writes, the formulation 2 of the mass matrix being employed :

$$J_{ij} = \begin{cases} \sum_{T \ni i} \left( \frac{m_{ii}^{F_2}}{\Delta t} + \beta_i^T k_i + \nu \frac{\|\mathbf{n}_i\|}{d^2 |T|} \right), & \text{if } i = j \\ \sum_{T \ni i,j} \left( \frac{m_{ij}^{F_2}}{\Delta t} + \beta_i^T k_j + \nu \frac{\mathbf{n}_i \cdot \mathbf{n}_j}{d^2 |T|} \right), & \text{if } i \neq j \end{cases} \quad (3.39)$$

### 3.1.5.3 Convergence Study

We perform a convergence study for the advection diffusion equation. An exact solution can be found for constant diffusion (see the study proposed in [61]), but only for constant advection speed. We have instead chosen to use a *manufactured solution* to solve (2.33). The principle of manufactured solutions, as explained in [142] is to define *a priori* the exact solution  $u_{ex}$  and to inject it into the considered equations. It will give a source term  $S_m$ . For equation (2.33) :

$$\frac{\partial u_{ex}}{\partial t} + \mathbf{a}(u_{ex}) \cdot \nabla u_{ex} - \nu \Delta u_{ex} = S_m$$

For the proposed computations (this study and as will be seen next section for the NS study), the source term is computed using the *Maxima* software. Then the original equation minus this source term is solved :

$$\frac{\partial u}{\partial t} + \mathbf{a}(u) \cdot \nabla u - \nu \Delta u - S_m = 0 \quad (3.40)$$

The manufactured solution is set such that derivative and second derivative are smooth :

$$u_{ex}(x, y, t) = \begin{cases} \cos^3(2\pi r(x, y, t)), & \text{if } r(x, y, t) \leq r_0 \\ 0, & \text{elsewhere} \end{cases} \quad (3.41)$$

with

$$r(x, y, t) = \sqrt{(x - \alpha t)^2 + y^2} \quad (3.42)$$

that corresponds with the "forced" advection of the cube cosinus at speed  $\alpha$ . This solution allows to perform computations not too costly, even on fine mesh and Dirichlet BC are used. The domain of computation is  $[-0.5, 0.7] \times [-0.5, 0.5]$ .  $\alpha$  and  $T_f$  are such that  $\alpha T_f = 0.2$ .

Two kinds of grids are employed. The first one is composed of a triangular grids whom triangles have been cut (see figure 3.1a) and is called "structured" through misuse of language. The second kind is totally unstructured and the grid are generated using the *gmsh* software (see figure 3.1b). So as to perform the convergence study, different mesh are considered such that the characteristic size  $h = \sqrt{\frac{L_x L_y}{N_v}}$  is divided by two from one mesh to another. Table 3.1 presents the number of nodes for the different meshes used and their associated length. The CFL is kept constant for all the simulations. For the RK2 scheme, it is set  $CFL = 1$  and for the CN scheme,  $CFL = 15$ . To invert the matrix, the MUMPS library is employed. The curves corresponding to the convergence are plotted figure 3.2a, 3.2b (structured, unstructured grids) for the  $N_1$  and  $N_2$  norms, defined as :

$$N_1 = \sum_{i=1}^{N_v} \frac{|u_{ex,i} - u_i|}{N_v}, \quad N_2 = \sum_{i=1}^{N_v} \sqrt{\frac{(u_{ex,i} - u_i)^2}{N_v}}$$

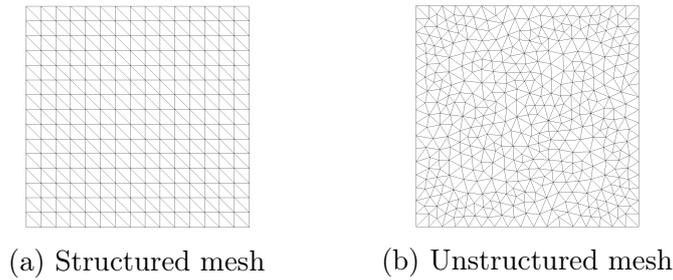


Figure 3.1 – Two kinds of grids for convergence study

$N_v$	1600	6400	25600	102400	409600
$h$	0.05	0.025	0.0125	0.00625	0.003125

Table 3.1 – Convergence Study : Number of nodes and characteristic length

with  $u_{ex,i}$  and  $u_i$  the exact and computed values of the solution at node  $i$ . The convergence is well recovered for both of the schemes. In addition, it is noticed that the Crank Nicolson scheme compares well with the RK2 scheme in terms of accuracy whereas the CFL condition is 15<sup>th</sup> time higher. Concerning the CPU time, the computations performed with the CN scheme are about twice faster than those performed with the RK scheme. This shows the time gain provided by implicit schemes, without degrading the precision of the solution. However, note that even if the CFL is set to 15 for the CN scheme, the computations are only twice faster (at best, the number of Newton iterations being dependent on the mesh size). This is in part due to the matrix inversion which is not optimal. In particular, the MUMPS library uses a direct method while in a Newton algorithm, a low tolerance in the matrix inversion is often enough. A better implementation with other matrix inversion algorithm would lead to better results, as will be seen for NS studies. Apart from this matrix inversion, better choices of Jacobian approximation may lead to better results.

### 3.1.6 Extension to Navier Stokes Equations

In this section, the proposed schemes are extended to the NS system of equations. The main points tackled here concern the temporal part of the problem, meaning expression of the mass matrix and the approximation of the Jacobian in the case of NS equations. The remaining part being identical to what has been discussed in section 2.3.4.1.

#### 3.1.6.1 Formulation of the mass matrix

We start by giving the formulations of the mass matrix for a system case. The extension is quite immediate. The mass matrix is now a tensor of size

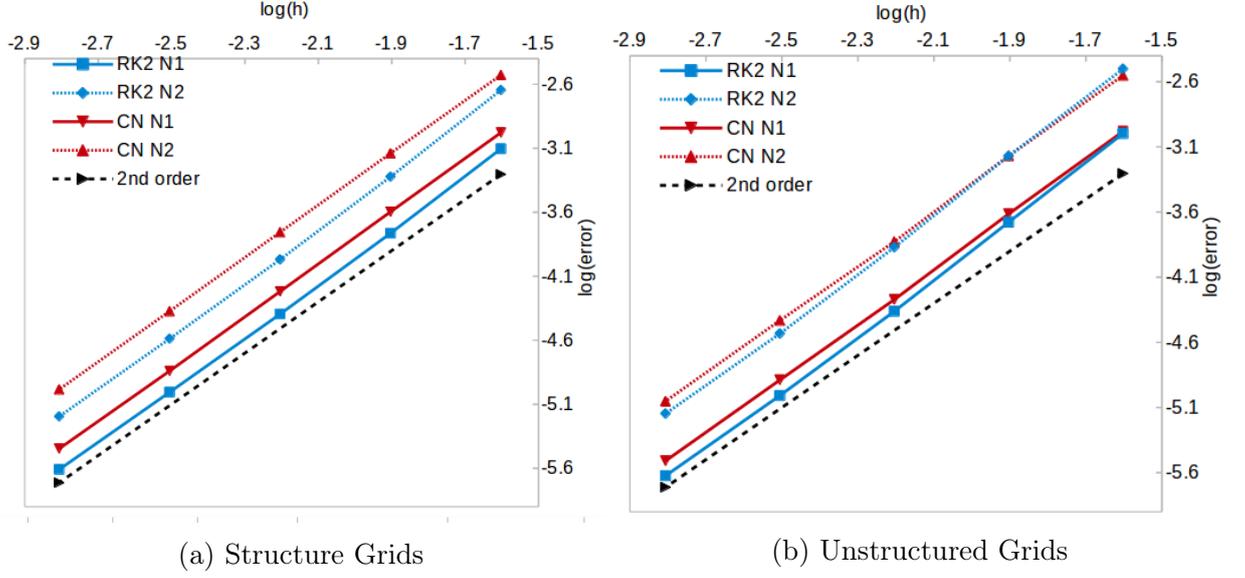


Figure 3.2 – Convergence Study - Unsteady Scalar Advection Diffusion

$N_{var} \times N_{var} \times N_{DoF} \times N_{DoF}$ . Thus, we denote by  $\underline{\mathbf{M}}_{ij}$  the matrix of size  $N_{var} \times N_{var}$ . The Formulation 1 writes :

$$\underline{\mathbf{M}}_{ij}^{F1} = \frac{|T|}{d+1} \underline{\boldsymbol{\beta}}_i^T \quad (3.43)$$

and the formulation 2 is defined by :

$$\underline{\mathbf{M}}_{ij}^{F2} = \frac{|T|}{d+1} \left( \frac{1 + \delta_{ij}}{d+2} \mathbb{I} + \underline{\boldsymbol{\beta}}_i^T - \frac{1}{d+1} \mathbb{I} \right) \quad (3.44)$$

Finally, the formulation (3.13) writes for a system of equation :

$$\Phi_i^T(u_h) = \sum_{j \in T} \underline{\mathbf{M}}_{ij} \frac{\Delta \mathbf{u}_j}{\Delta t} + \phi_i^T(\mathbf{u}_h) \quad (3.45)$$

It gives the following formulations for the schemes :

### 3.1.6.2 Second Order Explicit Runge Kutta Scheme

The formulation is just the extension of (3.25,3.26) to system using the mass matrix (3.43) :

$$\begin{cases} \mathbf{u}_i^* = \mathbf{u}_i^n + \sum_{T \ni i} \left[ \underline{\beta}_i^T \phi^T(\mathbf{u}_h^n) + \int_T \mathbb{K} \nabla \varphi_i \cdot (\nabla \mathbf{u}_h^n - \widetilde{\nabla \mathbf{u}_h^n}) \right] \\ \mathbf{u}_i^{n+1} = \mathbf{u}_i^* + \sum_{T \ni i} \left[ \sum_{j \in T} \underline{\mathbf{M}}_{ij}^{F1} \frac{\mathbf{u}_j^* - \mathbf{u}_j^n}{\Delta t} + \underline{\beta}_i^T \phi^T(\mathbf{u}_h^{n+\frac{1}{2}}) + \int_T \mathbb{K} \nabla \varphi_i \cdot (\nabla \mathbf{u}_h^{n+\frac{1}{2}} - \widetilde{\nabla \mathbf{u}_h^{n+\frac{1}{2}}}) \right] \end{cases} \quad (3.46)$$

with

$$\phi^T(\mathbf{u}_h) = \int_T (\nabla \cdot \mathbf{F}^{NS}(\mathbf{u}_h) - \nabla \cdot (\mathbb{K} \widetilde{\nabla \mathbf{u}_h^n})) \quad (3.47)$$

### 3.1.6.3 Implicit $\theta$ Scheme

Some additional informations are given here for the extension to the NS equations of the  $\theta$  scheme. The problem (3.31) writes :

$$\begin{cases} \underline{\mathbf{J}}(\mathbf{U}^{k+1} - \mathbf{U}^k) = -\mathbf{R}(\mathbf{U}^k) \\ \mathbf{U}^0 = \mathbf{U}^n \end{cases} \quad (3.48)$$

with

$$\mathbf{R}_i(\mathbf{u}_h) = \sum_{T \ni i} \left\{ \sum_{j \in T} \underline{\mathbf{M}}_{ij}^{F2} \frac{\mathbf{u}_j - \mathbf{u}_j^n}{\Delta t} + \phi_i^T(\mathbf{u}_h^{n+\theta}) \right\} \quad (3.49)$$

where this times,  $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_{N_v})^T$  and  $\mathbf{R} = (\mathbf{R}_1(\mathbf{u}_h), \dots, \mathbf{R}_{N_v}(\mathbf{u}_h))^T$  are of size  $N_v N_{var}$  and the matrix  $\underline{\mathbf{J}}$  is of size  $N_{var} N_v \times N_{var} N_v$ . The expression of the approximated Jacobian  $\underline{\mathbf{J}}_{ij}$  of size  $N_{var} \times N_{var}$  is :

$$\underline{\mathbf{J}}_{ij} = \begin{cases} \sum_{T \ni i} \left( \frac{\underline{\mathbf{M}}_{ii}^{F2}}{\Delta t} + \underline{\beta}_i^T \underline{\mathbf{K}}_i + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_i)}{d^2 |T|} \right), & \text{if } i = j \\ \sum_{T \ni (i,j)} \left( \frac{\underline{\mathbf{M}}_{ij}^{F2}}{\Delta t} + \underline{\beta}_i^T \underline{\mathbf{K}}_j + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_j)}{d^2 |T|} \right), & \text{if } i \neq j \end{cases} \quad (3.50)$$

However, for some complex simulations, this approximation was not sufficient and led to either too many Newton iterations, or to no convergence at all. In those situations, the Newton Algorithm has been slightly modified by using a dual time stepping (same principle than for steady simulation). It modifies problem (3.29) as follows :

$$\frac{\partial \mathbf{U}_h}{\partial \tau} + \mathbf{R}(\mathbf{U}_h) = 0$$

that is discretized with a simple implicit Euler scheme :

$$|C_i| \frac{\mathbf{U}_i^{k+1} - \mathbf{U}_i^k}{\Delta\tau} + \mathbf{R}_i(\mathbf{U}^{k+1}) = 0, \quad \forall i \in \mathcal{T}^h \quad (3.51)$$

where  $\Delta\tau$  is the dual time stepping. It modifies the original Jacobian by only reinforcing the diagonal and the new matrix  $\tilde{\mathbf{J}}$  to invert writes :

$$\tilde{\mathbf{J}}_{ij} = \begin{cases} \sum_{T \ni i} \left( \frac{|C_i|}{\Delta\tau^k} \mathbb{I} + \frac{\mathbf{M}_{ii}}{\Delta t} + \underline{\beta}_i^T \mathbf{K}_i + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_i)}{d^2 |T|} \right), & \text{if } i = j \\ \sum_{T \ni (i,j)} \left( \frac{\mathbf{M}_{ij}}{\Delta t} + \underline{\beta}_i^T \mathbf{K}_j + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_j)}{d^2 |T|} \right), & \text{if } i \neq j \end{cases} \quad (3.52)$$

$\Delta\tau^k$  denotes the fact that the dual time step is chosen according to the convergence of the method. The more it converges, the higher is this dual time step.

#### 3.1.6.4 Convergence study

As for the scalar equation in section 3.1.5.3, a manufactured solution is employed. The solution is chosen as follows for the pressure, velocity and density :

$$\begin{cases} \rho = \begin{cases} \rho_0 + \rho_0 \cos^3(2\pi r(x, y, t)), & \text{if } r \leq r_0 \\ \rho_0, & \text{elsewhere} \end{cases} \\ u = u_0 \\ v = \begin{cases} v_0 + v_0 \cos^3(2\pi r(x, y, t)), & \text{if } r \leq r_0 \\ v_0, & \text{elsewhere} \end{cases} \\ p = \begin{cases} p_0 + p_0 \cos^3(2\pi r(x, y, t)), & \text{if } r \leq r_0 \\ p_0, & \text{elsewhere} \end{cases} \end{cases} \quad (3.53)$$

with

$$r = \sqrt{(x - \alpha t)^2 + y^2} \quad (3.54)$$

It is the exact same problem than the scalar one, but for a system case. It corresponds to the 'forced' advection with speed  $\alpha$  of a cube cosinus on the density,  $v$  velocity and pressure. The computation is performed on a domain  $[-0.5, 0.7] \times [-0.5, 0.5]$  with  $\alpha T_f = 0.2$ . For the variables (3.53, 3.54), the following parameter are set :

$$\alpha = u_0 = 6, \quad v_0 = 1, \quad \rho_0 = 1.4, \quad p_0 = 100$$

Same grids than in section 3.1.5.3 are employed for the convergence study (see figures 3.1a, 3.1b and table 3.1). For sake of clarity, only the  $N_1$  norm of the

### 3.1. Residual Schemes for Unsteady Problems

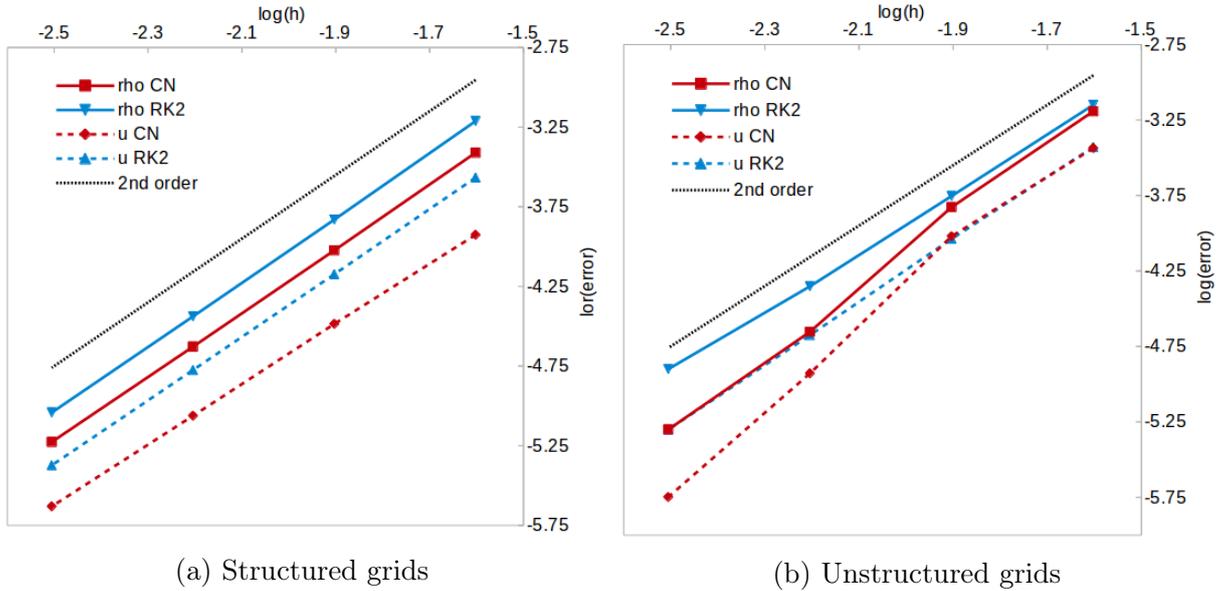


Figure 3.3 – Navier Stokes - Unsteady Convergence study,  $N_1$  norm

error of the density and  $u$  velocity are displayed figures 3.3a and 3.3b, but the results for  $v$  velocity and pressure are similar. A  $CFL = 1$  has been employed for the RK2 scheme and  $CFL = 15$  for the CN scheme. As expected, the second order accuracy is recovered for both schemes, which confirms the scalar study. Now, the focus is the time computation. During De Santis PhD [56], a Gmres algorithm with LU preconditioning has been implemented for the steady case. This provided matrix inversion has been used. Although the gain in time of computation is higher than for the scalar case (CN computation 3 times faster than RK2 one on the finest mesh), it is not optimal. However, we have preferred the simplicity provided by the proposed one, while choosing higher CFL for application test cases can still allow to decrease the computational time of 60/70%.

## 3.2 Extension to Penalization

### 3.2.1 A splitting approach

The approach employed for unsteady simulations is not the same than the one used for steady ones. It is based on a splitting, that will be detailed in the next section. This is motivated by the following points :

- A well known problem associated to penalization is that employed with an explicit scheme the time step should be of the same order than the penalty parameter. As this penalty parameter has to be chosen very small to ensure an accurate imposition of the BCs ( $\eta \sim 10^{-10}$ ), it leads to inconceivable simulations. However, when considering a splitting, the resolution of the NS part of the equation being "independent" of the penalization one, this constraint does not hold anymore and thus, explicit schemes can be employed.
- The present study falls within the continuity of previous works resolving penalization on cartesian grids for tracking solid motion into a fluid [26, 118]. In those references, an aerodynamical forces computation is proposed, based on a change of momentum computation specific for splitting approach (detailed section 3.2.4).
- This proposed approach does not involve any change in the Navier Stokes computation. Then, it is easily coupled to any already functional NS computational code.

### 3.2.2 The Strang Splitting

A splitting method consists into solving a PDE by solving it by part, leading to simpler PDEs. For instance, considering :

$$\partial_t u + a(\partial_{x^i}^i u, x, t) + b(\partial_{x^i}^i u, x, t) = 0 \quad (3.55)$$

with  $a$  and  $b$  arbitrary operations on the derivatives of  $u$  according to  $x$ . The two following equations are solved :

$$\partial_t u + a(\partial_{x^i}^i u, x, t) = 0 \quad (3.56)$$

$$\partial_t u + b(\partial_{x^i}^i u, x, t) = 0 \quad (3.57)$$

If  $M_{\Delta t}$  and  $N_{\Delta t}$  denotes respectively operators associated to the resolution of (3.56) and (3.57), a basic splitting consists in solving equation (3.55) by :

$$\begin{cases} u^1 = M_{\Delta t} u^n \\ u^{n+1} = N_{\Delta t} u^1 \end{cases} \quad (3.58)$$

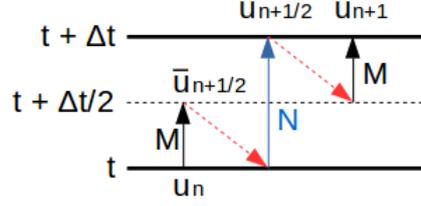


Figure 3.4 – Strang Splitting

The question of the accuracy is raised. Indeed, since the beginning, globally second order scheme have been proposed, and even if the penalization is only first order accurate, we compensate this loss by mesh adaptation. Thus, what would be required for the splitting to not spoil this accuracy?

Assuming  $M_{\Delta t}$  and  $N_{\Delta t}$  to be second order approximations (in time), their combination must be carefully chosen in order to preserve this accuracy. Strang has demonstrated [152] that the operator  $M_{\frac{\Delta t}{2}}(t + \frac{\Delta t}{2})N_{\Delta t}(t)M_{\frac{\Delta t}{2}}(t)$  provides the required accuracy. This process is sketched figure 3.4, and the notations that will be used are :

$$\begin{cases} \bar{u}^{n+\frac{1}{2}} = M_{\frac{\Delta t}{2}} u^n \\ u^{n+\frac{1}{2}} = N_{\Delta t} \bar{u}^{n+\frac{1}{2}} \\ u^{n+1} = M_{\frac{\Delta t}{2}} u^{n+\frac{1}{2}} \end{cases} \quad (3.59)$$

### 3.2.3 Application to penalization

#### 3.2.3.1 Development of the scheme

We now consider the following penalized equation :

$$\partial_t u + \nabla \cdot \mathcal{F}(u) - \nabla \cdot (\nu \nabla u) + \underbrace{\frac{1}{\eta}(u - u_S)}_{S(u)} = 0 \quad (3.60)$$

We are willing to apply splitting in order to solve separately :

$$\begin{cases} \partial_t u + \nabla \cdot \mathcal{F}(u) - \nabla \cdot (\nu \nabla u) = 0 \\ \partial_t u + \frac{1}{\eta}(u - u_S) = 0 \end{cases}$$

So as to ensure spatial consistency, the splitting is applied on the spatially discretized formulation of the equation. We first apply a RD scheme to this equation (3.60) :

$$\sum_{T \ni i} \left\{ \int_T \omega_i (\partial_t u_h + S(u_h)) + \phi_i^T(u_h) \right\} = 0 \quad (3.61)$$

For a linear approximation of the source term  $S$ , we have :

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij} (\partial_t u_j + S(u_j)) + \beta_i \phi^T(u_h) \right\} = 0 \quad (3.62)$$

We can now introduce the operators used in the splitting approach proposed:

$$\sum_{T \ni i} \sum_{j \in T} m_{ij} (\partial_t u_j + S(u_j)) = 0 \quad (3.63)$$

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij} \partial_t u_j + \beta_i \phi^T(u_h) \right\} = 0 \quad (3.64)$$

Under the assumption that matrix  $(m_{ij})$  is invertible, the first equation (3.63) admits a trivial pointwise solution

$$\partial_t u_i + S(u_i) = 0 \quad (3.65)$$

This approach is applied to the penalized NS system of equations. The penalization part becomes the system :

$$\begin{cases} \partial_t \rho_i = 0 \\ \partial_t (\rho_i \mathbf{u}_i) + \frac{1}{\eta} (\rho_i \mathbf{u}_i - \rho \mathbf{u}_{S_i}) = 0 \\ \partial_t (\rho_i e_i) + \frac{1}{\eta} (\rho \epsilon_{int,i} - \rho \epsilon_{intS,i}) + \frac{1}{\eta} (\rho_i \mathbf{u}_i - \rho \mathbf{u}_{S_i}) \cdot \mathbf{u}_i = 0 \end{cases} \quad (3.66)$$

System (3.66) can be simplified considerably by combining the ODEs involved. From the first equation, we get that  $\rho_i$  remains constant thus :

$$\partial_t \mathbf{u}_i + \frac{1}{\eta} (\mathbf{u}_i - \mathbf{u}_{S_i}) = 0 \quad (3.67)$$

Considering now the energy equation, it is easily obtained, using the definition of the total energy  $e_i = \epsilon_{int,i} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$ , and the previous equations :

$$\partial_t \epsilon_{int,i} + \frac{1}{\eta} (\epsilon_{int,i} - \epsilon_{intS,i}) + \mathbf{u}_i \cdot \underbrace{\left( \partial_t \mathbf{u}_i + \frac{1}{\eta} (\mathbf{u}_i - \mathbf{u}_{S_i}) \right)}_0 = 0 \quad (3.68)$$

Finally, we propose to use the following simplified version of (3.66)

$$\begin{cases} \partial_t \rho_i = 0 \\ \partial_t \mathbf{u}_i + \frac{1}{\eta} (\mathbf{u}_i - \mathbf{u}_{S_i}) = 0 \\ \partial_t \epsilon_{int,i} + \frac{1}{\eta} (\epsilon_{int,i} - \epsilon_{intS,i}) = 0 \end{cases} \quad (3.69)$$

All the penalized variables are now uncoupled, and satisfy an ODE of the type:

$$\partial_t f + \frac{1}{\eta}(f - f_S) = 0 \quad (3.70)$$

Equation (3.64) will now provide the RDS discretization of the standard NS equations (2.1).

In the following, we denote by  $S_{(P,\Delta t)}$  the operator associated to the penalization system (3.69), and by  $S_{(NS,\Delta t)}$  the operator resulting from the RD discretization of the NS equations (2.1). To get a global second order accuracy, a second order RDS in space and time and a second order operator  $S_{(P,\Delta t)}$  have to be chosen. For the NS part, the schemes proposed previously section 3.1.6 can be employed (explicit schemes being no longer restrained by the CFL constraint). The next two following paragraphs present the approach used to solve (3.70).

### 3.2.3.2 Constant penalty variable

We have to solve over an interval  $[0, \Delta t]$

$$\begin{cases} \partial_t f + \frac{1}{\eta}(f - f_S) = 0 \\ f(0) = f_{IC} \end{cases} \quad (3.71)$$

where  $f_{IC}$  is the initial condition. The exact solution of (3.71) with  $f_S$  constant is :

$$f(t) = f_{IC}e^{-\frac{t}{\eta}} + (1 - e^{-\frac{t}{\eta}})f_S \quad (3.72)$$

thus the operator for (3.65) which is exact is :

$$S_{(P,\frac{\Delta t}{2})} = f_{IC}e^{-\frac{\Delta t}{2\eta}} + (1 - e^{-\frac{\Delta t}{2\eta}})f_S \quad (3.73)$$

and the Strang splitting applied to NS - (3.69) writes :

$$\begin{cases} \bar{\mathbf{U}}^{n+\frac{1}{2}} = \mathbf{U}^n e^{-\frac{\Delta t}{2\eta}} + (1 - e^{-\frac{\Delta t}{2\eta}})\mathbf{U}_S \\ \mathbf{U}^{n+\frac{1}{2}} = S_{(NS,\Delta t)}\bar{\mathbf{U}}^{n+\frac{1}{2}} \\ \mathbf{U}^{n+1} = \mathbf{U}^{n+\frac{1}{2}} e^{-\frac{\Delta t}{2\eta}} + (1 - e^{-\frac{\Delta t}{2\eta}})\mathbf{U}_S \end{cases} \quad (3.74)$$

with the first and last step only applied to  $\mathbf{u}$  and  $\epsilon_{int}$ .

### 3.2.3.3 Time dependant penalty variable

To handle the case in which  $f_S$  is an arbitrary function of time, we propose to compute a particular solution to (3.71) introducing the asymptotic expansion w.r.t. the small paramter  $\eta$  :

$$f = f_0 + \eta f_1 + \eta^2 f_2 + \dots$$

Replacing this expression in the original ODE, and equating equal powers of  $\eta$  we readily find the asymptotic particular solution :

$$f = f_S + \sum_i (-\eta)^i \partial_t^i f_S \quad (3.75)$$

But at this point, the context of penalization intervenes. As  $\eta$  is set very small ( $10^{-10}$ ), every term  $\eta^i, i \geq 2$  is neglected. So, to an accuracy of order  $\eta^2$ , we consider the particular solution:

$$f_{par}(t) \simeq f_S(t) - \eta \partial_t f_S \quad (3.76)$$

And thus the truncated exact solution on the interval  $[0, \Delta t]$  becomes :

$$f(t) = (f_{IC} - f_S(0) + \eta \partial_t f_S) e^{-\frac{t}{\eta}} + f_S(t) - \eta \partial_t f_S \quad (3.77)$$

It is easily seen that for constant variables of penalization the derivative parts are cancelled, and (3.72) is found back. If  $\partial_t f_S$  is known analytically, its expression can be explicitly used. Otherwise, some discrete approximation of this quantity can also be employed. In our cases, the operators for the first and last operation of the Strang splitting can be computed by using (3.77) with  $f_{IC} = f^n$  and  $f_{IC} = f^{n+\frac{1}{2}}$  :

$$\bar{f}^{n+\frac{1}{2}} = f^n e^{-\frac{\Delta t}{2\eta}} + (\eta \partial_t f_S^n - f_S^n) e^{-\frac{\Delta t}{2\eta}} + f_S^{n+\frac{1}{2}} - \eta \partial_t f_S^{n+\frac{1}{2}} \quad (3.78)$$

$$f^{n+1} = f^{n+\frac{1}{2}} e^{-\frac{\Delta t}{2\eta}} + \left( \eta \partial_t f_S^{n+\frac{1}{2}} - f_S^{n+\frac{1}{2}} \right) e^{-\frac{\Delta t}{2\eta}} + f_S^{n+1} - \eta \partial_t f_S^{n+1} \quad (3.79)$$

## 3.2.4 Fluid Structure Interaction : Evaluation of the Aerodynamic Forces

Two different approaches are studied to compute the aerodynamical forces exerted on the solid. The first one, proposed by Morency *et al* in [26, 118] is specifically designed for splitting method. The second one is the classical way of computed forces by integrating the shear stress and the pressure on the surface.

### 3.2.4.1 Change of Momentum Computation

As said previously, this computation is specific for splitting techniques. Indeed, the idea is to evaluate the change of momentum in a close surface/volume in  $2D/3D$  containing the body. However, when using splitting, during the Navier Stokes part, the solid is "not seen" and thus the volume on which the integration is performed is the solid itself. This integral computation being performed between the NS step of the splitting and the penalization one. From an implementation point of view, this technique is quiet simple to settle. Indeed, the integration is performed over the elements on the solid, that are already marked for the penalization step. The change of momentum is performed as :

$$\Delta \mathbf{m} = \int_S \rho(\mathbf{u} - \mathbf{u}_s) = \sum_{T \in S} \int_T \rho(\mathbf{u} - \mathbf{u}_s) \quad (3.80)$$

and the forces is computed as :

$$\mathbf{F} = \frac{\Delta \mathbf{m}}{\Delta t} \quad (3.81)$$

Identically, the pitching moment is computed as :

$$\mathbf{T} = \frac{1}{\Delta t} \int_S \mathbf{r} \times \rho(\mathbf{u} - \mathbf{u}_s) \quad (3.82)$$

with  $\mathbf{r}$  the lever arm. In the following, the forces computed using this method will be denoted  $\mathbf{F}_{CM}$ .

### 3.2.4.2 Surface Integral Computation

The second way considered to compute the forces is by integrating the pressure and shear stress contribution on the surface. However, as the surface is not known explicitly, the first step is to interpolate the solution from the computational domain on a discretization of the surface and to then perform the integration. Thus, the necessary quantity (pressure and gradients) are recovered on the discretized surface using a simple  $P_1$  interpolation and the integration is performed over the elements  $\partial T$  of the surfacic mesh :

$$\mathbf{F} = \int_S (-p\mathbb{I} + \mathbb{S})\mathbf{n}_{norm}dS = \sum_{\partial T} \int_{\partial T} (-p\mathbb{I} + \mathbb{S})\mathbf{n}_{norm}dS \quad (3.83)$$

where  $\mathbf{n}_{norm}$  is the normalized normal of the solid. Identically :

$$\mathbf{T} = \int_S \mathbf{r} \times (-p\mathbb{I} + \mathbb{S})\mathbf{n}_{norm}dS \quad (3.84)$$

**Remark 3.3.** The interpolation step is not costly. So as to locate on which element of the computational mesh a node of the surface discretization belongs, at each time step, the starting element is the containing element at the previous time step which is very close to the researched one.

From now on, the forces computed using this technique will be denoted  $\mathbf{F}_{IC}$ .

### 3.2.5 Results

To validate the approach proposed, several test cases are considered. The first ones are the Rayleigh test cases corresponding to the study of the flow put into motion by a wall, with constant or oscillating speed. This study proposes the comparison of the obtained solution with an analytical one. The second case corresponds to the study of the evolution of a flow past a 2D cylinder. This problem is very well documented in the literature and allows to validate both the method and the aerodynamical forces computations.

**Remark 3.4.** This test case is not strictly speaking a motionless body as the velocity is not null. However, we employ "motionless" to characterize bodies defined by a fixed level set on the mesh.

#### 3.2.5.1 Rayleigh test cases

The Rayleigh Problem consists in the development of the flow of a motionless fluid which is dragged by a wall moving at speed (cf. figure 3.5a)

$$u(t) = U \cos(\omega t)$$

with  $\omega$  the frequency. For a constant speed ( $\omega = 0$ ), the analytical solution for this problem is, for an incompressible flow [146] :

$$u(y, t) = U \operatorname{erf} \left( \frac{y}{2\sqrt{\nu t}} \right)$$

where  $U$  is the wall speed,  $\nu$  is the kinematic viscosity and  $t$  the time.

For an incompressible fluid, an analytical solution can be found also in the case of an oscillating wall, and it is given by [146] :

$$u(y, t) = U e^{-ky} \cos(ky - \omega t), \quad k = \sqrt{\frac{\omega}{2\nu}}$$

The computations are performed from  $t = 0$  to  $T_f = 0.2$  for the constant speed case, and from  $t = 0$  to  $T_f = 0.46875$  for an oscillating wall. In the latter case we have taken  $\omega T_f = \frac{5\pi}{2}$  so as to have more than one period for the simulation time. The dynamic viscosity is set to  $\mu = \nu\rho = 0.1$ .

The computational domain is in both cases  $[0, 1] \times [0, 1]$  for non penalized simulations, and  $[0, 1] \times [-0.2, 1]$  for the penalized ones. The penalty term is imposed on the band  $[0, 1] \times [-0.2, 0]$  (see figure 3.5b). The left and right boundaries are periodic, and the free stream condition on the top is given by :  $(\rho, u, v, p) = (1.4, 0, 0, 100)$ .

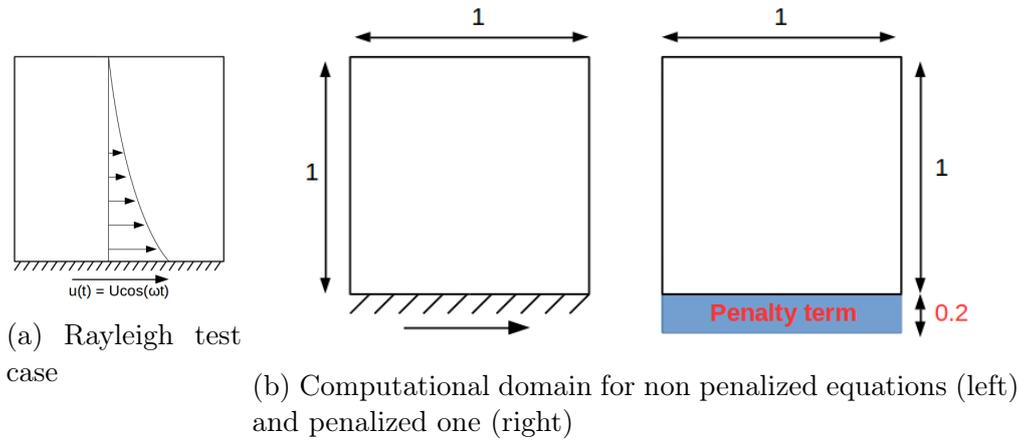


Figure 3.5 – Rayleigh test case presentation

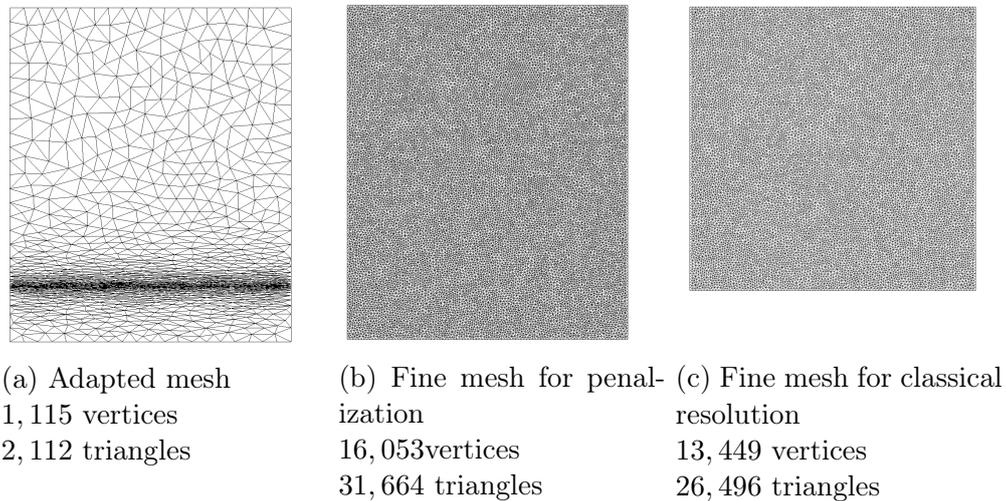


Figure 3.6 – Rayleigh tests - Different meshes

We will compare the solutions obtained on a mesh adapted w.r.t. the level set, and on two uniform fine meshes. One for a penalized simulation and the other for a fitted one. As for steady simulations, the mesh is adapted to the 0 level set value and is generated using as adaptation parameters  $\epsilon = 2.10^{-3}$ ,  $h_{min} = 2, 5.10^{-3}$  and  $h_{max} = 7, 5.10^{-2}$ . The three meshes are presented figures 3.6a, 3.6b and 3.6c.

The velocity profiles at  $x = 0.5$ , along with a zoom close to the wall, are plotted figure 3.7 using penalization on the adapted and fine meshes compared to the solution of the classical NS equations on a fine mesh and the analytical solution. It appears that the penalized solution is as good as the classical solution. Both solutions are in good agreement with the analytical one. The solution on the adapted mesh compares well to the solutions obtained on fine

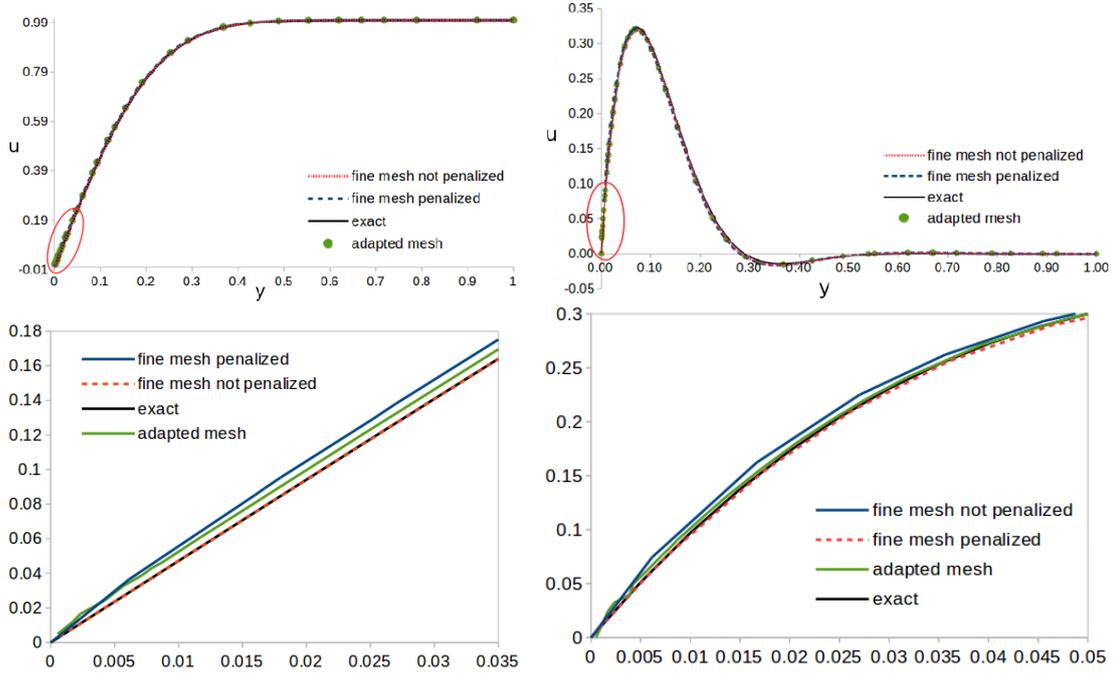


Figure 3.7 – Speed profile on different grids. Left : Constant speed - Right : Oscillating speed. Top : General - Bottom : Zoom close to the wall

meshes, which proves once again that mesh adaptation is a powerful tool : same accuracy for more than ten times less vertices. In addition, looking at the isolines of the velocity plotted figure 3.8, we can see that mesh adaptation allows to improve the accuracy of the solution at the fluid/solid interface, and thus to better impose the wanted BCs. The solution close to the interface plotted figure 3.7 shows that even if the penalization does not perfectly match with the fitted simulation and analytical solution, the results are still improved thanks to the adaptation.

### 3.2.5.2 Flow Past Cylinder

In this section, we consider the development of a von Karman vortex street behind a circular cylinder. We will compare the forces obtained with and without penalization, and the literature. The lift and drag coefficients (respectively  $C_L$  and  $C_D$ ) are derived from the aerodynamical forces  $F_x, F_y$  by :

$$\begin{cases} C_L = \frac{F_y}{1/2\rho_\infty u_\infty^2 L} \\ C_D = \frac{F_x}{1/2\rho_\infty u_\infty^2 L} \end{cases} \quad (3.85)$$

where  $L$  is the characteristic length of the solid,  $\rho_\infty$  and  $u_\infty$  the inflow density and velocity. The cylinder (radius 0.5) centered in  $(0, 0)$  is located in a

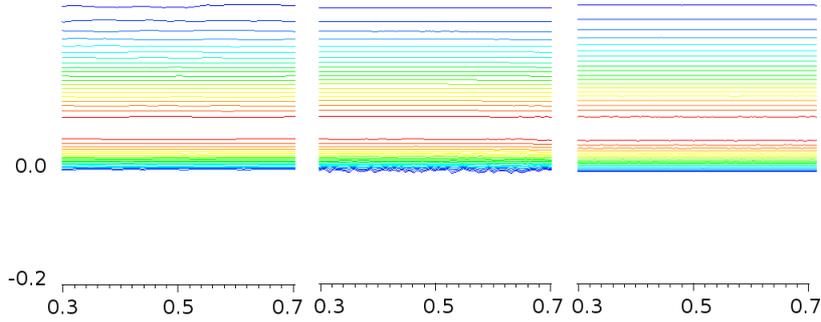


Figure 3.8 – Speed isoline. Left : Adapted mesh with penalization - Middle : Fine mesh with penalization (poor definition of the solid) - Right : Fine mesh without penalization

rectangular domain  $[-6, 10] \times [-12, 12]$  meshed with different mesh sizes (figure 3.9a). This choice has been done so as to avoid boundary effect, to have acceptable boundary treatment and to not dissipate the Von Karman alleys. The Reynolds is set to  $Re = 200$ . Inflow is defined by  $Ma = 0.2$ ,  $\rho = 1$  and a pressure of  $1/\gamma$ .

Two different meshes are generated for this test case. The first one contains an explicit discretization of the solid. It allows to perform the computation of the aerodynamical forces  $\mathbf{F}_{IC}$  without interpolation. In addition, there is no loss of accuracy because of the geometry approximation, and the two forces computation can then be compared for an exact geometry. The second mesh is an adapted one, on which the force is only computed with the change of momentum. Figures 3.9b and 3.9c present the meshes close to the 0 level set function. The adapted mesh was obtained from the fitted one with the following parameters :  $\epsilon = 0.001$ ,  $h_{min} = 0.0025$ . The fitted grid contains 67815 vertices, 135456 elements and the adapted one 85945 vertices and 171715 elements.

Plots of the lift and drag coefficients for the different meshes with the different methods are presented figure 3.10. The first noticeable result is that the computation of the lift coefficient is almost identical whatever is the mesh and the way of computing the forces (the maximum deviation is of 1.6%).

The mean value of the drag coefficient and the Strouhal number are compared with literature results. Bergmann *et al.* in [29] performed a study for the control and the optimization of the drag coefficients. They compared their results with literature. Thus, we use their work as a base for comparison. We add our values to the table presented in [29], see table 3.2. For all the computations, the Strouhal number is of 0.1965 which is really close to the literature results.

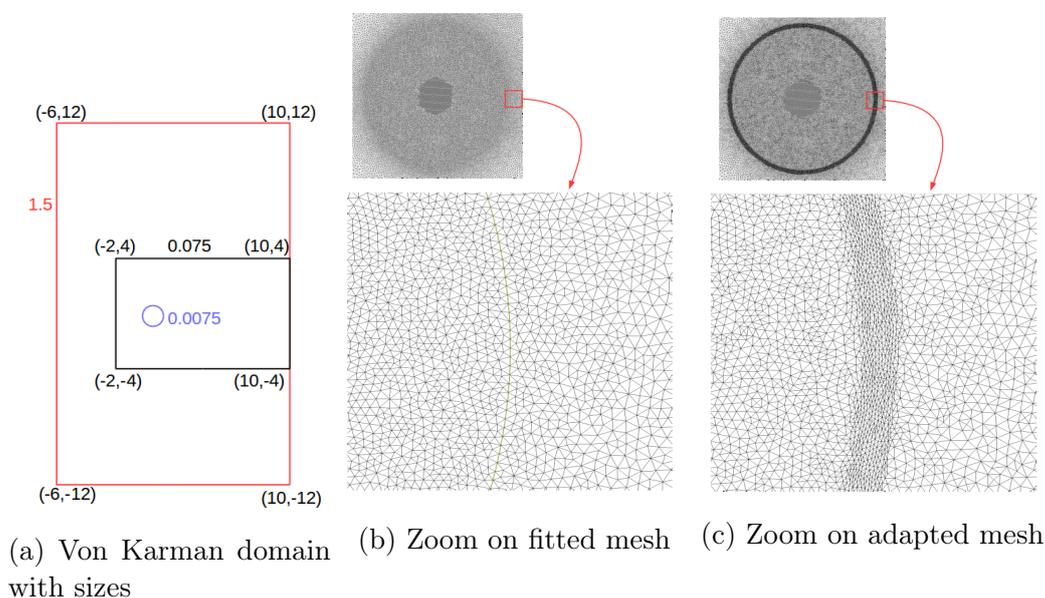


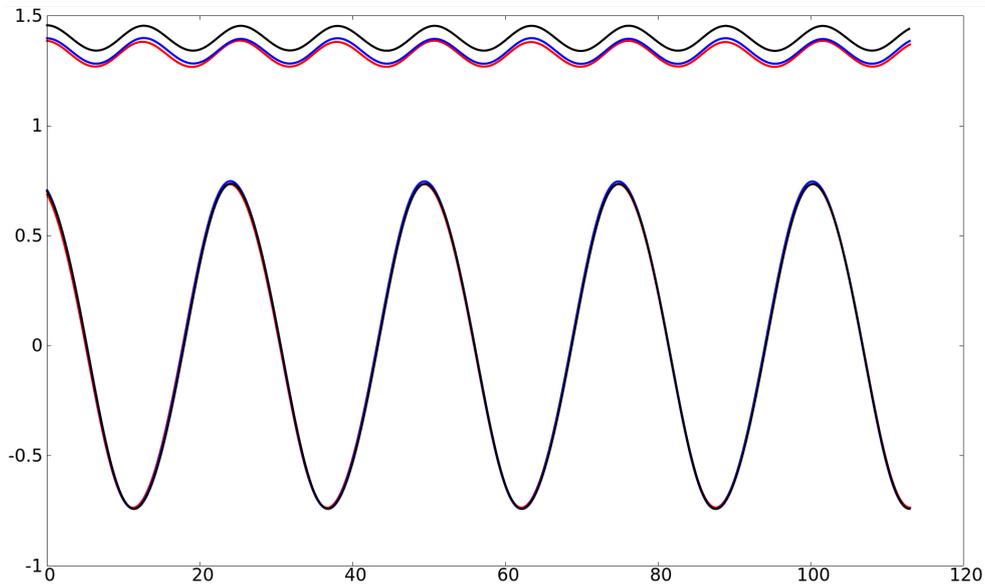
Figure 3.9 – Flow past cylinder

Now, we first look at the result with the "classical" force computation on fitted mesh (integration over the edges of the pressure and shear stress). The mean value is in good agreement with the other given mean values. This, added to the good Strouhal number and lift coefficient allow us to confirm the validation of our penalized method solved with RD scheme and splitting.

Then, we focus on the results given by the change of momentum computation. For both meshes, the mean values are a little under the others. This is an expected result because this way of computing is not as precise as the direct integration on the solid boundaries. However, looking more in detail (see figure 3.11), we can see some little fluctuations in the maximum value of the sine curves that have not been explained. Nevertheless, we compute the forces so as to perform simulations with bodies moved by the fluids, and as these fluctuations induces (on the adapted mesh) an error of about 0.4%, we consider it irrelevant according to our aim. Thus, we can validate our force computation as well as the proposed method. In addition, we notice that the difference on the drag coefficient between adapted and fitted mesh is of 0.9% which proves that mesh adaptation allow to recover the same results than the ones computed on fitted mesh.

To highlight the importance of performing correctly the gradient reconstruction, we propose three computations : the resolution of the classical NS equations on a fitted mesh; two simulations with penalization with and without the gradient correction (2.111), on a mesh obtained from the fitted

Author	St	$C_D$
Braza <i>et al.</i> [33]	0.2000	1.4000
Henderson [84]	0.1971	1.3412
He <i>et al.</i> [83]	0.1978	1.3560
Bergmann <i>et al.</i> [29]	0.1999	1.3900
Fitted Mesh, "classical computation"	0.1965	1.3979
Fitted Mesh, "change of mass computation"	0.1965	1.3404
Adapted Mesh	0.1965	1.3280

Table 3.2 – Strouhal Number and mean drag coefficient value for  $Re = 200$ Figure 3.10 – Lift and Drag Coefficient for  $Re = 200$ . Black : Fitted Mesh "classical computation" - Blue : Fitted Mesh "change of mass computation" - Red : Adapted Mesh

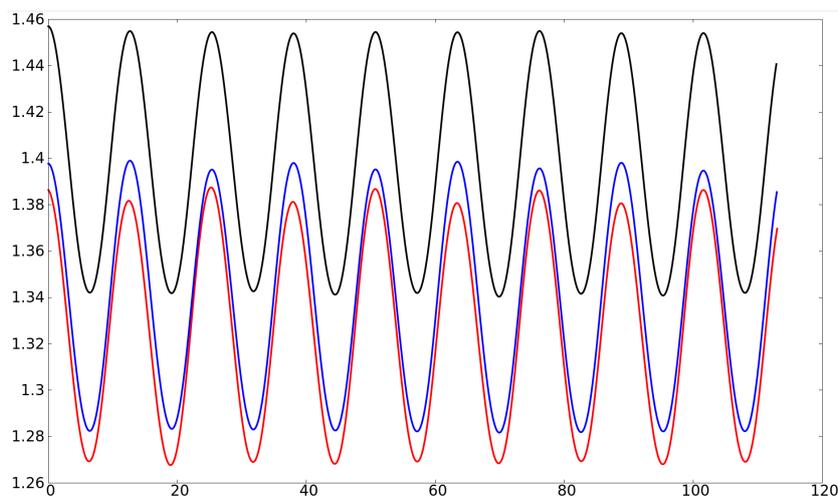


Figure 3.11 – Drag coefficient for  $Re = 200$ . Black : Fitted Mesh "classical computation" - Blue : Fitted Mesh "change of mass computation" - Red : Adapted Mesh

one by meshing the inside of the cylinder. The fitted mesh contains 34210 vertices and 68140 elements, while the mesh including the cylinder contains 39018 nodes and 77454 triangles. We compute the drag coefficients by integrating the pressure and shear stress. The results are plotted on figure 3.12. We also plot the  $\tau_{xy}$  component of the stress tensor along the segment  $[AB]$ ,  $A(-0.5, 0.25)B(0.5, -0.25)$  on figure 3.13. The curves clearly show that without modifications, the shear stresses computed by the RD method are wrong in vicinity of solid walls. The modification proposed allows to recover the correct value of these quantities.

### 3.3 Summary

In this chapter, the time dependent penalized Navier Stokes equations have been solved using a splitting approach. Such a strategy provides some flexibility in the choice of the scheme employed for the Navier Stokes resolution, explicit ones being no longer constrained by the penalty parameter. The approach has thus been validated using a RK2 RD scheme, that has been considered in a first time for its efficient implementation. A CN RD scheme has also been developed, allowing to increase the CFL and reduce the time of computation. Both of the schemes are an extension to unsteady problems of the high order schemes proposed by De Santis [56], with a restriction to second order accuracy. The approach has been validated on test cases, along with forces computations, that are required to be accurate to perform FSI simulations. Now that a generic method is validated to solve unsteady penalized problems, numerical tools are required to handle moving objects cases, that is the focus

of the next section.

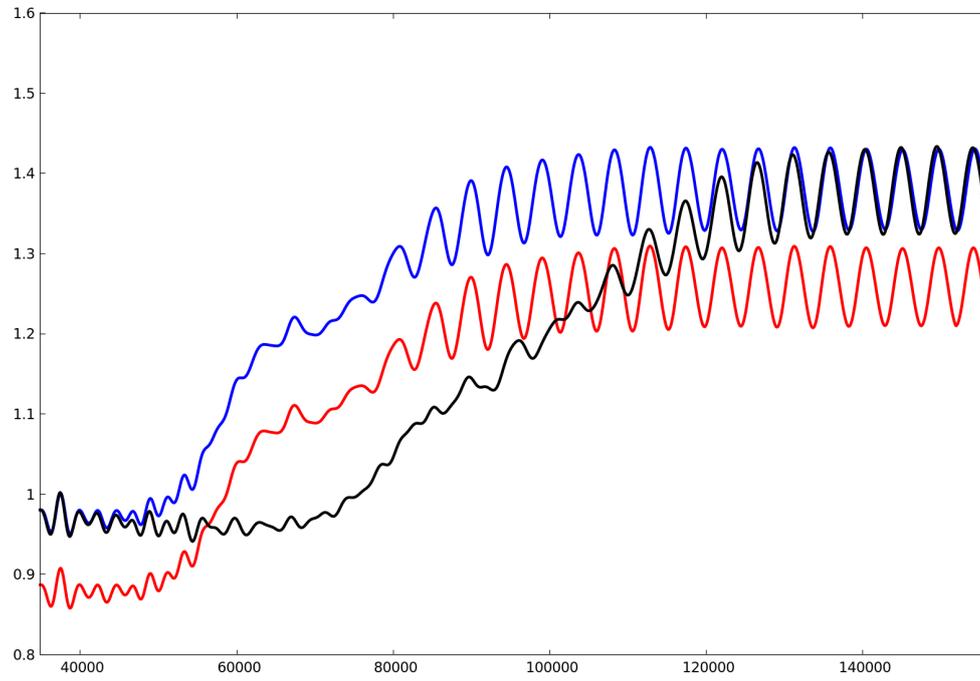


Figure 3.12 – Computation of drag coefficients for different simulations : Classical simulation (black), penalized simulation with new gradient reconstruction (blue) and without (red)

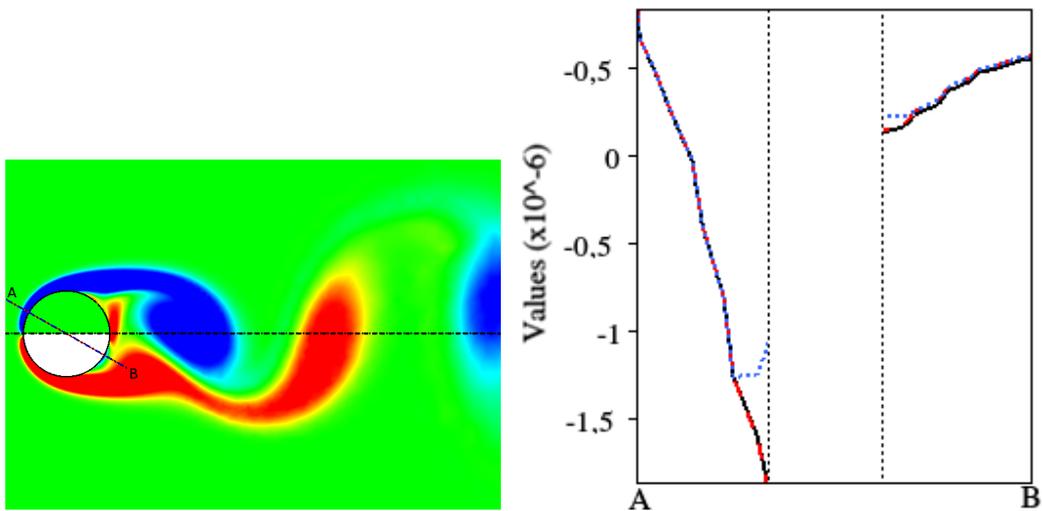


Figure 3.13 – Flow past cylinder computations - Left : vorticity (top : penalized simulation, bottom : classical simulation) and cutting segment  $[AB]$  for plotting stress - Right : Plot of the  $\tau_{xy}$  component of the stress tensor, fitted computation (black), penalization with new gradient reconstruction (dashed red) and without (dotted blue).



## Chapter 4

# Penalization and Moving Bodies - An ALE Approach

Some works have been provided to extend the metric based mesh adaptation strategy presented chapter 2 to unsteady problems [94, 81, 28]. The process differs in Jannoun's PhD [94] from the one adopted by Alauzet and co authors [81, 28], but the basic idea is the same : the generation of an optimal mesh for a given time interval (this time interval being optimal in [94]). In [94], simulations of windmills and rotating helicopter propeller have been performed by employing an IB method. However, even if the results are quite convincing, the remeshing/interpolation steps required when switching from one time interval to the next one motivated us to look at other alternatives. That is why we orientated the research on the combination of r-adaptivity techniques and Arbitrary Lagrangian Eulerian simulations.

In a first section, a recalling of the integral formulation of conservation laws in the ALE framework is proposed, and the schemes previously presented are expressed into this specific context. A non exhaustive state of the art about some r-adaptivity techniques is then provided, and the approach used here is described. A final section discusses the process used to solve the penalization on adaptive mesh, with academic test cases validating the methodology proposed, and an application to the trajectory of an ice shape.

## 4.1 Residual Distribution Scheme For Solving Arbitrary Lagrangian Eulerian Problems

### 4.1.1 State of the art

In the Lagrangian framework, the nodes of the computational domain are associated to the material particles and follow their motion across time. Thus the mesh is deformed according to the motion of the flow. The main weakness of this approach is then that in presence of large distortion or vorticity, mesh tangling can be obtained and a remeshing phase is required. However, its power resides in its ability to easily track surfaces, interfaces or discontinuities. On the opposite, the Eulerian approach considers a fixed computational mesh, and the motion is performed with respect to the grid. Thus this approach can deal with large distortion, but reduces the possibility to track precisely interfaces. The ALE approach aims at combining the two previous ones, to get the advantages of both. The mesh is mobile, but independent of the particles motion, which allows to perform refinement without distortions. It consists in a reformulation of the conservation laws (see for instance [64, 20, 60, 72]) that takes into account the mesh velocity.

This kind of approach dates back from the end of the 60s, especially with the work of Hirt *et. al.* [85]. The notion of Discrete Geometric Conservation Law (DGCL), ensuring that the geometric parameters are computed such that a uniform solution is kept constant, whatever is the mesh deformation, has then been introduced (see for instance [107, 79, 67]). Especially, in [107], Lesoine and Farhat proposed an imposition of the DGCL at a middle configuration between the mesh at time  $t^n$  and time  $t^{n+1}$ . Another approach with an imposition of the DGCL at time  $t^{n+1}$  can also be considered (see for instance [78]). One can see the chapter proposed by Donea *et. al.* [64] in *Encyclopedia of Computational Mechanics* and the references therein for a more exhaustive presentation and different applications, in fluid mechanics but also non linear solid mechanics.

In more direct relation with the present work, RD schemes has been proposed for ALE Euler simulation in Dobes PhD [78], where the moving mesh contribution in the equations is treated as a source term, and more recently Arpaia *et. al.* proposed a second order ALE explicit Runge Kutta scheme where the DGCL proposed in [107] is employed [20]. In [68], Farhat and Lakshminarayan employ ALE formalism to perform simulation using embedded boundary methods. Indeed, it allows to keep a refined area by translating/rotating it close to the implicit definition of the solid, but for small displacements.

## 4.1.2 Conservation Laws In ALE Formulation

This section presents the fundamental aspects of the *Arbitrary Lagrangian Eulerian* approach for CFD.

### 4.1.2.1 Generalities and notations

#### Referential, Lagrangian and Spatial domain

To express the ALE equations, three configurations are considered : the *Lagrangian* or *material* one, denoted by  $\Omega_{\mathbf{X}}$ , the *spatial* configuration, on which we are willing to express the conservation law, denoted by  $\Omega_{\mathbf{x}}$ , that will move with respect to a *referential* configuration, denoted by  $\Omega_{\boldsymbol{\chi}}$ . The motion leading to the spatial configuration is totally independent from the material one.

The coordinates are expressed as  $\mathbf{X} = (X, Y)$  in the material domain,  $\mathbf{x} = (x, y)$  in the spatial one and  $\boldsymbol{\chi} = (\chi, \xi)$  in the referential configuration. We denote by  $\Phi$  the transformation from the referential domain to the spatial one, by  $\varphi$  the one from the material domain to the spatial one, and by  $\Psi$  the mapping from the referential configuration to the material one (see figure 4.1). Obviously, whatever is the considered transformation, as two points in a domain can not coincide to the same mapping in the image domain,  $\Phi$ ,  $\varphi$  and  $\Psi$  are diffeomorphisms. Thus, letting :

$$\begin{aligned} \Phi : \Omega_{\boldsymbol{\chi}} \times \mathbb{R}^+ &\rightarrow \Omega_{\mathbf{x}} \times \mathbb{R}^+ \\ (\boldsymbol{\chi}, t) &\rightarrow \Phi(\boldsymbol{\chi}, t) = (\mathbf{x}, t) \end{aligned}$$

if consider the Jacobian  $\mathcal{J}_{\Phi} = \partial \mathbf{x} / \partial \boldsymbol{\chi}$ , and  $J_{\Phi} = \det(\mathcal{J}_{\Phi})$ , we have  $J_{\Phi} \neq 0$ . The *domain velocity*  $\boldsymbol{\sigma}$  can be defined as :

$$\boldsymbol{\sigma} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\boldsymbol{\chi}} \quad (4.1)$$

Identically, considering the mapping from the material to the spatial configuration :

$$\begin{aligned} \varphi : \Omega_{\mathbf{X}} \times \mathbb{R}^+ &\rightarrow \Omega_{\mathbf{x}} \times \mathbb{R}^+ \\ (\mathbf{X}, t) &\rightarrow \varphi(\mathbf{X}, t) = (\mathbf{x}, t) \end{aligned}$$

the Jacobian is denoted  $\mathcal{F}_{\varphi} = \partial \mathbf{x} / \partial \mathbf{X}$  and as previously we have :  $\det(\mathcal{F}_{\varphi}) = J_{\varphi} \neq 0$ . The *material velocity*  $\mathbf{a}$  is defined by :

$$\mathbf{a} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{X}} \quad (4.2)$$

Considering the link between the referential and material domains, as expressed in [64], it is convenient to consider the inverse  $\Psi^{-1}$  of  $\Psi$  (which is

well defined as we consider a diffeomorphism) instead of  $\Psi$  itself :

$$\begin{aligned}\Psi^{-1} : \Omega_{\mathbf{X}} &\rightarrow \Omega_{\boldsymbol{\chi}} \\ (\mathbf{X}, t) &\rightarrow \Psi^{-1}(\mathbf{X}, t) = (\boldsymbol{\chi}, t)\end{aligned}$$

and the *particle velocity* in the referential domain  $\mathbf{w}$  is expressed as :

$$\mathbf{w} = \left. \frac{\partial \boldsymbol{\chi}}{\partial t} \right|_{\mathbf{x}} \quad (4.3)$$

it corresponds with the variation of the referential coordinates  $\boldsymbol{\chi}$  linked to the particle located at  $\mathbf{X}$  and keeping it fixed. It is easily seen that  $\varphi = \Phi \circ \Psi^{-1}$ , and the relation between the domain velocity (4.1), the material velocity (4.2) and the particle velocity (4.3) is obtained by :

$$\begin{aligned}\mathbf{a} &= \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{x}} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\boldsymbol{\chi}} + \left. \frac{\partial \boldsymbol{\chi}}{\partial t} \right|_{\mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \boldsymbol{\chi}} \\ \mathbf{a} &= \boldsymbol{\sigma} + \mathbf{w} \cdot \frac{\partial \mathbf{x}}{\partial \boldsymbol{\chi}}\end{aligned}$$

This last relation allows to define  $\mathbf{v}$ , the relative velocity between the material domain and the spatial one. This velocity is called *convective velocity* :

$$\mathbf{v} = \mathbf{a} - \boldsymbol{\sigma} = \mathbf{w} \cdot \frac{\partial \mathbf{x}}{\partial \boldsymbol{\chi}} \quad (4.4)$$

From now on, derivatives of quantities will be considered. To ease the reading, the following notations will be used to distinguish the material derivative and the spatial derivative :

$$\begin{aligned}\text{material derivative : } & \left. \frac{\partial \cdot}{\partial t} \right|_{\mathbf{x}} = \frac{d \cdot}{dt} \\ \text{spatial derivative : } & \left. \frac{\partial \cdot}{\partial t} \right|_{\mathbf{x}} = \frac{\partial \cdot}{\partial t}\end{aligned}$$

#### 4.1.2.2 ALE Integral form of a conservation law

This section aims at express in an integral form a conservation law in the ALE framework as done in *e.g.* [72, 20]. The starting point is to express in the spatial referential the integral conservation of a quantity  $u$  on an arbitrary volume  $C(t)$  :

$$\frac{d}{dt} \int_{C(t)} u(\mathbf{x}, t) dV = 0 \quad (4.5)$$

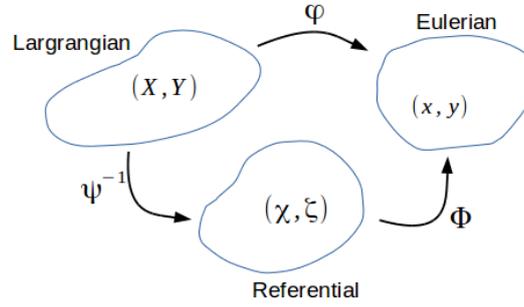


Figure 4.1 – ALE formalism : Referential, Lagrangian and Eulerian configurations

In order to propose an integral formulation, relations linking the different derivatives of a quantity  $u$  are given :

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u \quad (4.6)$$

$$\left. \frac{\partial u}{\partial t} \right|_{\mathbf{x}} = \frac{\partial u}{\partial t} + \boldsymbol{\sigma} \cdot \nabla u \quad (4.7)$$

In addition of those two relations, a property that will be called *Geometric Conservation Law* (GCL) needs to be expressed. It imposes via equalities considering the Jacobian determinants  $J_\varphi$  and  $J_\Phi$  some constraints concerning the spatial deformation and guaranteeing the conservation of volume.

**Property 4.1.** *Geometric Conservation Law*

$$\begin{aligned} \left. \frac{\partial J_\Phi}{\partial t} \right|_{\mathbf{x}} &= J_\Phi \nabla \cdot \boldsymbol{\sigma} \\ \frac{dJ_\varphi}{dt} &= J_\varphi \nabla \cdot \mathbf{a} \end{aligned} \quad (4.8)$$

*Proof :*

The proof is proposed for the first relation, an analogue reasoning leading to the second one. We start from

$$J_\Phi = \frac{\partial x}{\partial \chi} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \chi}$$

and then

$$\begin{aligned}
 \left. \frac{\partial J_\Phi}{\partial t} \right|_{\mathbf{x}} &= \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} \left( \frac{\partial x}{\partial \chi} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \chi} \right) \\
 &= \frac{\partial x}{\partial \chi} \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} \frac{\partial y}{\partial \xi} + \frac{\partial y}{\partial \xi} \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} \frac{\partial x}{\partial \chi} - \frac{\partial x}{\partial \xi} \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} \frac{\partial y}{\partial \chi} - \frac{\partial y}{\partial \chi} \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} \frac{\partial x}{\partial \xi} \\
 &= \frac{\partial x}{\partial \chi} \frac{\partial \sigma_y}{\partial \xi} + \frac{\partial y}{\partial \xi} \frac{\partial \sigma_x}{\partial \chi} - \frac{\partial x}{\partial \xi} \frac{\partial \sigma_y}{\partial \chi} - \frac{\partial y}{\partial \chi} \frac{\partial \sigma_x}{\partial \xi} \\
 &= \frac{\partial x}{\partial \chi} \frac{\partial y}{\partial \xi} \partial_y \sigma_y + \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \chi} \partial_x \sigma_x - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \chi} \partial_y \sigma_y - \frac{\partial y}{\partial \chi} \frac{\partial x}{\partial \xi} \partial_x \sigma_x \\
 &= \left( \frac{\partial x}{\partial \chi} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \chi} \right) (\partial_x \sigma_x + \partial_y \sigma_y) \\
 &= J_\Phi \nabla \cdot \boldsymbol{\sigma} \quad \square
 \end{aligned}$$

Now that the GCL is established, two integral relations can be expressed. First, considering the volume  $C(t)$  as the image of  $C_X$  by the mapping  $\varphi$ , it can be written :

$$\begin{aligned}
 \frac{d}{dt} \int_{C(t)} u dV &= \int_{C_X} \frac{d(J_\Phi u)}{dt} dV \\
 &= \int_{C_X} \left( J_\Phi \frac{du}{dt} + u \frac{dJ_\Phi}{dt} \right) dV \\
 &= \int_{C_X} J_\Phi \left( \frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u + u \nabla \cdot \mathbf{a} \right) dV \text{ using GCL 4.1 and (4.6)} \\
 &= \int_{C(t)} \left( \frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u + u \nabla \cdot \mathbf{a} \right) dV \\
 &= \int_{C(t)} \left( \frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{a}u) \right) dV
 \end{aligned}$$

which is in fact an alternative form of the Reynolds transport theorem. Expressing  $\mathbf{a}u$  in a more generic form as the flux  $\mathcal{F}$  of the conserved quantity  $u$ , we can write :

$$\frac{d}{dt} \int_{C(t)} u dV = \int_{C(t)} \left( \frac{\partial u}{\partial t} + \nabla \cdot \mathcal{F} \right) dV = 0 \quad (4.9)$$

**Remark 4.1.** It can be noticed here that if  $C(t)$  remains constant in time, we find back the Eulerian formulation of the conservation law :  $\partial_t u + \nabla \cdot \mathcal{F} = 0$ .

Secondly, we are considering the volume  $C(t)$  as the mapping in the spatial domain of a volume  $C_x$  in the referential one *via* the transformation  $\varphi$ , and the following variation is considered :

$$\begin{aligned}
 \frac{\partial}{\partial t} \Big|_{\mathbf{x}} \int_{C(t)} u dV &= \int_{C_x} \frac{\partial(J_\varphi u)}{\partial t} \Big|_{\mathbf{x}} dV \\
 &= \int_{C_x} \left( J_\varphi \frac{\partial u}{\partial t} \Big|_{\mathbf{x}} + u \frac{\partial J_\varphi}{\partial t} \Big|_{\mathbf{x}} \right) dV \\
 &= \int_{C_x} J_\varphi \left( \frac{\partial u}{\partial t} + \boldsymbol{\sigma} \nabla u + u \nabla \cdot \boldsymbol{\sigma} \right) dV \text{ using GCL 4.1 and (4.7)} \\
 &= \int_{C(t)} \left( \frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\sigma} u) \right) dV
 \end{aligned}$$

From which it is written :

$$\int_{C(t)} \frac{\partial u}{\partial t} = \frac{\partial}{\partial t} \Big|_{\mathbf{x}} \int_{C(t)} u dV - \int_{C(t)} \nabla \cdot (\boldsymbol{\sigma} u) dV \quad (4.10)$$

Using this relation (4.10) into the equality (4.9) leads to the integral ALE formulation of the conservation law :

$$\frac{\partial}{\partial t} \Big|_{\mathbf{x}} \int_{C(t)} u dv + \int_{C(t)} \nabla \cdot (\mathcal{F} - \boldsymbol{\sigma} u) dV = 0 \quad (4.11)$$

#### 4.1.2.3 Discrete Geometric Conservation Law

Now, the concept of Discrete Geometric Conservation Law (DGCL) needs to be introduced. It is commonly seen as the fact that the computations of the geometric parameters must allow to keep constant a uniform solution, whatever is the mesh motion [107, 79, 67]. It is in fact a requirement on the scheme to be consistent with the volume conservation expressed at the continuous level by the GCL. In this work, following the work of [20, 60] for ALE RD schemes, based on the idea introduced in [107] for FV scheme, the choice is to impose the DGCL using the configuration at time  $t^{n+\frac{1}{2}}$ . Thus, considering discretization (3.1) for a one step scheme (as a simple explicit Euler discretization) applied to an ALE integral formulation of an advection diffusion conservation law, we obtain :

$$\int_{T^{n+1}} \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \frac{u_h^n}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \nabla \cdot (\mathcal{F}^n - \mathcal{G}^n - \boldsymbol{\sigma} u^n) = 0 \quad (4.12)$$

As the DGCL imposes the scheme to keep a constant solution uniform, considering  $u$  constant in (4.12), the following DGCL is sorted out :

$$\int_{T^{n+\frac{1}{2}}} \nabla \cdot \boldsymbol{\sigma} = \frac{|T^{n+1}| - |T^n|}{\Delta t} \quad (4.13)$$

and any one step time discretization lead to the same formulation. Considering second order accuracy, it leads to ( $\sigma$  is piecewise linear,  $\nabla \cdot \sigma$  is constant by element) :

$$\nabla \cdot \sigma = \frac{|T^{n+1}| - |T^n|}{|T^{n+\frac{1}{2}}| \Delta t} \quad (4.14)$$

that leads to an explicit value of  $\sigma$  at each node  $i$  of the mesh :

$$\sigma_i = \frac{\mathbf{x}_i^{n+1} - \mathbf{x}_i^n}{\Delta t} \quad (4.15)$$

**Remark 4.2.** This nodal definition of the values of  $\sigma$  can only be obtained from the choice of the  $T^{n+\frac{1}{2}}$  configuration. For other choices, nodal definition may not be found, and one has to go through the definition of  $\nabla \cdot \sigma$  or of  $\sigma$  on the edges.

### 4.1.3 RD Schemes for scalar conservation law in ALE Form

This section is devoted to the reformulation of the schemes introduced in the previous section (3.1) in an ALE framework. First, the general formulation of a RD scheme for a one step time discretization is proposed for a scalar advection equation. The extension to advection diffusion problem is immediate. The specific schemes employed (RK2 RD and CN RD schemes) are then formulated. Finally, a convergence study is performed so as to verify the proposed approach.

### 4.1.4 General Formulation

To obtain ALE formulation of RD schemes, we consider the PG analogy (3.10). The Galerkin part  $\mathcal{L}_h^G$ , and the stabilized one  $\mathcal{L}_h^S$  are considered separately. The Galerkin operator is defined by discretizing the conservation law (4.11) as :

$$\mathcal{L}_h^G = \sum_T \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \varphi_i \nabla \cdot (\mathcal{F} - \sigma u_h) \right\} \quad (4.16)$$

The stabilization operator can be defined in the same form. However, as proposed in [20], a non conservative formulation of the equation for the "ALE Flux" is used, defining the operator as :

$$\mathcal{L}_h^S = \sum_T \left\{ \int_{T^{n+\frac{1}{2}}} \gamma_i \frac{\Delta u_h}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \gamma_i (\nabla \cdot \mathcal{F} - \sigma \cdot \nabla u_h) \right\} \quad (4.17)$$

This form (4.17) still provide a conservative scheme as  $\sum_{i \in T} \gamma_i = 0$  and satisfies the DGCL condition. Combining (4.16) and (4.17), the general formulation

for a one step ALE-RD scheme writes :

$$\sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i (u_h \nabla \cdot \boldsymbol{\sigma}) + \int_{T^{n+\frac{1}{2}}} \gamma_i \frac{\Delta u_h}{\Delta t} + \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(u_h) \right\} = 0 \quad (4.18)$$

where  $\widetilde{\phi}_i^T(u_h)$  stands for the ALE nodal fluctuation and is expressed as :

$$\widetilde{\phi}_i^T(u_h) = \int_T \omega_i (\nabla \cdot \mathcal{F} - \boldsymbol{\sigma} \cdot \nabla u_h) \quad (4.19)$$

that is rewritten in the RD formalism :

$$\widetilde{\phi}_i^T(u_h) = \beta_i^T \int_T (\nabla \cdot \mathcal{F} - \boldsymbol{\sigma} \cdot \nabla u_h) \quad (4.20)$$

When accounting for the viscous flux, the ALE nodal residual is modified using the reconstructed gradient and the penalization issued from the FOS that is consistent with the DGCL. Thus, for an advection diffusion conservation law the ALE nodal fluctuation (4.20) writes :

$$\widetilde{\phi}_i^T(u_h) = \beta_i^T \int_T \left[ \nabla \cdot \mathcal{F} - \boldsymbol{\sigma} \cdot \nabla u_h - \nabla \cdot (\underline{\mathbf{K}} \widetilde{\nabla} u_h) \right] + \int_T \underline{\mathbf{K}} \nabla \varphi_i \cdot (\nabla u_h - \widetilde{\nabla} u_h) \quad (4.21)$$

and the ALE fluctuation  $\widetilde{\phi}^T$  then writes :

$$\widetilde{\phi}^T(u_h) = \int_T \left[ \nabla \cdot \mathcal{F} - \boldsymbol{\sigma} \cdot \nabla u_h - \nabla \cdot (\underline{\mathbf{K}} \widetilde{\nabla} u_h) \right] \quad (4.22)$$

From this ALE fluctuation (4.22), we propose the definition of the inflow parameter (2.57) in the ALE framework, denoted  $k_i^\sigma$  :

$$k_i^\sigma = \frac{1}{d} (\bar{\mathbf{a}} - \bar{\boldsymbol{\sigma}}) \cdot \mathbf{n}_i \quad (4.23)$$

**Remark 4.3.** The distribution coefficient  $\beta_i^T$  used to distribute the ALE fluctuation have to be defined with respect to the ALE formulation. For instance, the SUPG distribution coefficient is computed with the ALE inflow parameters :

$$\beta_i^T = \frac{1}{N_{DoF}} + k_i^\sigma \tau$$

The two next sections illustrate how the RK2 and CN schemes are extended to the ALE framework. A geometrical relation holding for second order accuracy, allowing to simplify the expressions need to be recalled first. Indeed, coordinates  $\mathbf{x}$  and normals  $\mathbf{n}$  are linear and thus, the configuration at time  $t^{n+\frac{1}{2}}$  can be defined exactly by :

$$\mathbf{x}^{n+\frac{1}{2}} = \frac{\mathbf{x}^{n+1} + \mathbf{x}^n}{2}, \quad \mathbf{n}^{n+\frac{1}{2}} = \frac{\mathbf{n}^{n+1} + \mathbf{n}^n}{2}$$

However, the same relation does not hold for the areas. Nevertheless, for second order accuracy, as proposed in Dobeš' PhD [60], we consider :

$$|T^{n+\frac{1}{2}}| \approx \frac{|T^{n+1}| + |T^n|}{2} \quad (4.24)$$

#### 4.1.4.1 Explicit RK2 Scheme

The RK2 scheme (3.25) presented section 3.1.5.1 has been formulated for ALE problem by Arpaia *et. al.* in [20]. We briefly recall here the construction, considering directly the advection diffusion equation case.

Considering the operator associated to the stabilization (4.17), the same Time Shifted Operator (TSO) than introduced section 3.1.5.1 are employed :

$$\mathcal{L}_h^S = \sum_T \left\{ \int_{T^{n+\frac{1}{2}}} \gamma_i \frac{\widetilde{\Delta u_h}}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \gamma_i [\nabla \cdot (\mathcal{F} - \mathcal{G}) - \sigma \cdot \nabla u_h] \right\} \quad (4.25)$$

Thus, the first step writes :

$$\sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^*}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i (u_h^n \nabla \cdot \sigma) + \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^n) \right\} = 0 \quad (4.26)$$

Applying mass lumping on the two first terms and using (4.13), it gives :

$$\frac{|C_i^{n+1}|}{\Delta t} u_i^* - \frac{|C_i^n|}{\Delta t} u_i^n + \sum_{T \ni i} \left\{ -\frac{|T^{n+1}| - |T^n|}{\Delta t |T^{n+\frac{1}{2}}|} \int_{T^{n+\frac{1}{2}}} \varphi_i u_h^n + \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^n) \right\} = 0$$

And mass lumping can be applied again considering the following :

$$\sum_{T \ni i} \frac{|T^{n+1}| - |T^n|}{\Delta t |T^{n+\frac{1}{2}}|} \int_{T^{n+\frac{1}{2}}} \varphi_i u_h^n \approx \frac{|C_i^{n+1}| - |C_i^n|}{\Delta t} u_i^n \quad (4.27)$$

This finally leads to the first step of the ALE RK2 RD scheme :

$$\frac{|C_i^{n+1}|}{\Delta t} (u_i^* - u_i^n) + \sum_{T \ni i} \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^n) = 0 \quad (4.28)$$

For the second step, using the TSO (3.20) in the stabilized part (4.25) leads to :

$$\begin{aligned} \sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i (u_h^{n+\frac{1}{2}} \nabla \cdot \sigma) \right. \\ \left. + \int_{T^{n+\frac{1}{2}}} \gamma_i \frac{u_h^* - u_h^n}{\Delta t} + \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^{n+\frac{1}{2}}) \right\} = 0 \end{aligned}$$

As in section 3.1.5.1, the Galerkin integral  $\int_{T^{n+\frac{1}{2}}} \varphi_i \frac{u_h^* - u_h^n}{\Delta t}$  is added/subtracted :

$$\begin{aligned} & \sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i \left( u_h^{n+\frac{1}{2}} \nabla \cdot \boldsymbol{\sigma} \right) \right. \\ & \left. - \int_{T^{n+\frac{1}{2}}} \varphi_i \frac{u_h^* - u_h^n}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \omega_i \frac{u_h^* - u_h^n}{\Delta t} + \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^{n+\frac{1}{2}}) \right\} = 0 \end{aligned} \quad (4.29)$$

Now, using (4.24), mass lumping on the Galerkin integrals, and (4.27), the final expression of the second step of the ALE RK2 RD scheme writes :

$$\frac{|C_i^{n+1}|}{\Delta t} (u_i^{n+1} - u_i^*) + \sum_{T \ni i} \left( \sum_{j \in T} m_{ij} \frac{u_j^* - u_j^n}{\Delta t} + \widetilde{\phi_i^{T^{n+\frac{1}{2}}}}(u_h^{n+\frac{1}{2}}) \right) = 0 \quad (4.30)$$

#### 4.1.4.2 Implicit $\theta$ Scheme

The  $\theta$  scheme combined with the LDA RD scheme has been studied in [60] for advection problems. Here, the scheme is proposed in a general RD formalism and studied for advection diffusion. The Galerkin operator (4.16) writes :

$$\mathcal{L}_h^G = \sum_T \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \varphi_i \nabla \cdot (\mathcal{F} - \mathcal{G} - \boldsymbol{\sigma} u_h^{n+\theta}) \right\} \quad (4.31)$$

For the stabilized one (4.17), the whole equation is considered on the fixed middle geometry  $T^{n+\frac{1}{2}}$  :

$$\mathcal{L}_h^S = \sum_T \left\{ \int_{T^{n+\frac{1}{2}}} \gamma_i \frac{u_h^{n+1} - u_h^n}{\Delta t} + \int_{T^{n+\frac{1}{2}}} \gamma_i \nabla \cdot [(\mathcal{F} - \mathcal{G}) - \boldsymbol{\sigma} \cdot \nabla u_h^{n+\theta}] \right\} \quad (4.32)$$

As for the second step of the RK scheme, we add/subtract the Galerkin integral  $\int_{T^{n+\frac{1}{2}}} \varphi_i \frac{u_h^{n+1} - u_h^n}{\Delta t}$  :

$$\begin{aligned} & \sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{u_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{u_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i (u_h^{n+\theta} \nabla \cdot \boldsymbol{\sigma}) - \int_{T^{n+\frac{1}{2}}} \varphi_i \frac{u_h^{n+1} - u_h^n}{\Delta t} + \right. \\ & \left. \int_{T^{n+\frac{1}{2}}} \omega_i \left( \frac{u_h^{n+1} - u_h^n}{\Delta t} + \nabla \cdot (\mathcal{F}^{n+\theta} - \mathcal{G}^{n+\theta}) - \boldsymbol{\sigma} \cdot \nabla u_h^{n+\theta} \right) \right\} = 0 \end{aligned} \quad (4.33)$$

It appears, using mass lumping, (4.27) and (4.24), that the sum of the Galerkin integrals is in fact null for  $\theta = \frac{1}{2}$ , which gives the Crank-Nicolson scheme.

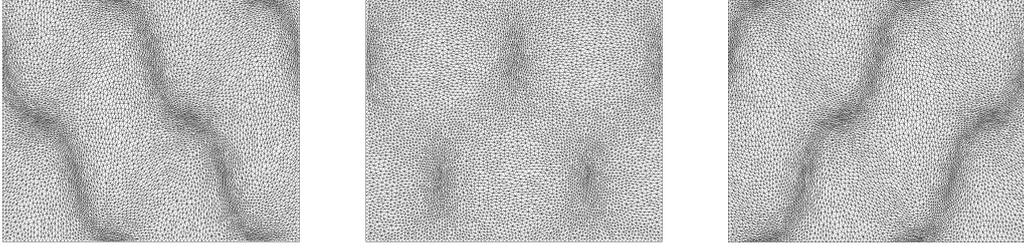


Figure 4.2 – Mesh Deformation for convergence study

Considering the summation (multiplied by  $\Delta t$ ) :

$$\begin{aligned} & \sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i u_h^{n+1} - \int_{T^n} \varphi_i u_h^n - \frac{|T^{n+1}| - |T^n|}{|T^{n+\frac{1}{2}}|} \int_{T^{n+\frac{1}{2}}} \varphi_i \frac{u_h^{n+1} + u_h^n}{2} - \int_{T^{n+\frac{1}{2}}} \varphi_i (u_h^{n+1} - u_h^n) \right\} \\ &= |C_i^{n+1}| u_i^{n+1} - |C_i^n| u_i^n - (|C_i^{n+1}| - |C_i^n|) \frac{u_i^{n+1} + u_i^n}{2} - \frac{|C_i^{n+1}| + |C_i^n|}{2} (u_i^{n+1} - u_i^n) \\ &= 0 \end{aligned}$$

Thus, the ALE CN scheme writes in the same form than for non ALE conservation law, except that the integrals are evaluated on the middle configuration  $T^{n+\frac{1}{2}}$ , and the ALE inflow parameter  $k_i^\sigma$  (4.23) is employed :

$$\sum_{T \ni i} \left\{ \sum_{j \in T} m_{ij} \frac{u_j^{n+1} - u_j^n}{\Delta t} + \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(u_h^{n+\frac{1}{2}}) \right\} = 0 \quad (4.34)$$

Thus, for the resolution, the exact same Newton algorithm than proposed section 3.1.5.2 is employed, with the same approximation in the Jacobian.

#### 4.1.4.3 Convergence Study

The convergence study is performed on the same test case than section 3.1.5.3, with an imposed motion of the mesh. The domain is  $[-0.5, 0.7] \times [-0.5, 0.5]$  and the motion is ruled by (see figure 4.2) :

$$\begin{cases} x = X + 0.05 \sin \left( 4\pi \frac{X + 0.5}{1.2} \right) \sin(2\pi(Y + 0.5)) \sin(4\pi(t + \Delta t)) \\ y = Y + 0.1 \sin \left( 4\pi \frac{X + 0.5}{1.2} \right) \sin(2\pi(Y + 0.5)) \sin(8\pi(t + \Delta t)) \end{cases} \quad (4.35)$$

The convergence study is performed on the same grids than for the non ALE study (see figure 3.1a and 3.1b for the two kinds of mesh and table 3.1 for the different sizes employed). The CFL employed are  $CFL = 0.8$  for the RK2 scheme (the mesh being deformed, even if the time step computation takes into account the nodes displacement, it ensures to stay in the stability

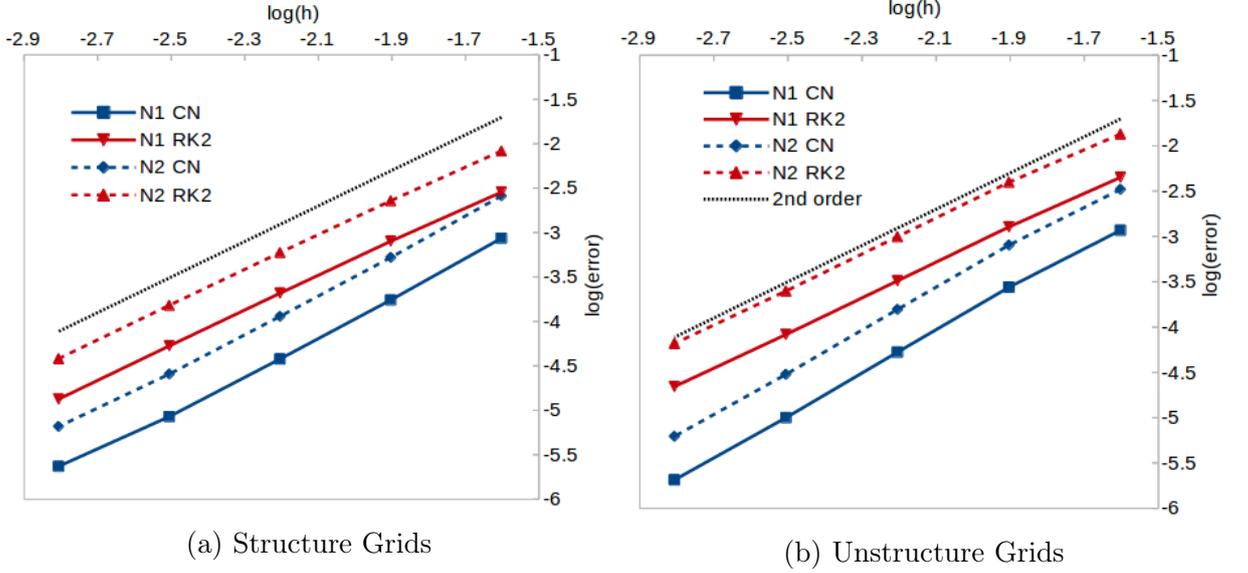


Figure 4.3 – Convergence study - ALE Scalar advection diffusion

zone), and  $CFL = 30$  for the CN scheme. The convergence curves are plotted figures 4.3a and 4.3b. Once again, the expected second order accuracy is well recovered for both ALE RK2 RD and ALE CN RD schemes. In addition, as previously, the time saving by using an implicit scheme is emphasized here. Looking at the curves, the accuracy compares well, but the computational time is this time approximatively 10 time higher for the RK2 scheme. Nevertheless, the same comment than for non ALE simulations can be done, a CFL almost 40 times higher leads to a computational time "only" 10 times higher.

## 4.1.5 Extension To Navier Stokes equations

### 4.1.5.1 Formulation of the schemes

This section is dedicated to give some details on the extension of the previous schemes to system case. The general formulation (4.18) of a one step scheme writes now :

$$\left\{ \begin{array}{l} \sum_{T \ni i} \left\{ \int_{T^{n+1}} \varphi_i \frac{\mathbf{u}_h^{n+1}}{\Delta t} - \int_{T^n} \varphi_i \frac{\mathbf{u}_h^n}{\Delta t} - \int_{T^{n+\frac{1}{2}}} \varphi_i (\mathbf{u}_h \nabla \cdot \boldsymbol{\sigma}) + \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(\mathbf{u}_h) \right\} = 0 \\ \widetilde{\phi}_i^T(\mathbf{u}_h) = \underline{\beta}_i^T \widetilde{\phi}^T(\mathbf{u}_h) + \int_T \mathbb{K} \nabla \varphi_i \cdot (\nabla \mathbf{u}_h - \widetilde{\nabla} \mathbf{u}_h) \end{array} \right. \quad (4.36)$$

with the ALE fluctuation :

$$\widetilde{\phi}^T(\mathbf{u}_h) = \int_T \left[ \nabla \cdot \mathbf{F}^{Eul}(\mathbf{u}_h) - \boldsymbol{\sigma} \cdot \nabla \mathbf{u}_h - \nabla \cdot (\mathbb{K} \widetilde{\nabla} \mathbf{u}_h) \right] \quad (4.37)$$

The ALE inflow matrices are defined, as for the scalar case by :

$$\underline{\mathbf{K}}_i^\sigma = \frac{1}{d} \sum_{k=1}^d (\mathbb{A}_k(u_h) - \sigma_k \mathbb{I}) n_{i_k} \quad (4.38)$$

The considered schemes can then be written for an application to the NS system of equations.

#### ALE RK2 RD Scheme

$$\begin{cases} \frac{|C_i^{n+1}|}{\Delta t} (\mathbf{u}_i^* - \mathbf{u}_i^n) + \sum_{T \ni i} \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(\mathbf{u}_h^n) = 0 \\ \frac{|C_i^{n+1}|}{\Delta t} (\mathbf{u}_i^{n+1} - \mathbf{u}_i^*) + \sum_{T \ni i} \left( \sum_{j \in T} \underline{\mathbf{M}}_{ij}^{F_1} \frac{\mathbf{u}_j^* - \mathbf{u}_j^n}{\Delta t} + \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(\mathbf{u}_h^{n+\frac{1}{2}}) \right) = 0 \end{cases} \quad (4.39)$$

#### ALE $\theta$ RD Scheme

The system to solve is :

$$\begin{cases} \underline{\mathbf{J}}(\mathbf{U}^{k+1} - \mathbf{U}^k) = -\mathbf{R}(\mathbf{U}^k) \\ \mathbf{U}^0 = \mathbf{U}^n \end{cases} \quad (4.40)$$

with

$$\mathbf{R}_i(\mathbf{u}_h) = \sum_{T \ni i} \left\{ \sum_{j \in T} \underline{\mathbf{M}}_{ij}^{F_2} \frac{\mathbf{u}_j - \mathbf{u}_j^n}{\Delta t} + \widetilde{\phi}_i^{T^{n+\frac{1}{2}}}(\mathbf{u}_h^{n+\theta}) \right\} \quad (4.41)$$

and the approximated Jacobian matrix for the Newton iterations writes :

$$\tilde{\underline{\mathbf{J}}}_{ij} = \begin{cases} \sum_{T \ni i} \left( \frac{|C_i|}{\Delta \tau^k} \mathbb{I} + \frac{\underline{\mathbf{M}}_{ii}^{F_2}}{\Delta t} + \underline{\beta}_i^T \underline{\mathbf{K}}_i^\sigma + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_i)}{d^2 |T|} \right), & \text{if } i = j \\ \sum_{T \ni (i,j)} \left( \frac{\underline{\mathbf{M}}_{ij}^{F_2}}{\Delta t} + \underline{\beta}_i^T \underline{\mathbf{K}}_j^\sigma + \frac{\mathbf{n}_i \cdot (\mathbb{K} \mathbf{n}_j)}{d^2 |T|} \right), & \text{if } i \neq j \end{cases} \quad (4.42)$$

#### 4.1.5.2 Convergence Study

The same convergence study than in section 3.1.6.4 is performed, using the mesh deformation proposed section 4.1.4.3, ruled by equations (4.35). For sake of clarity, as in section 3.1.6.4, only the  $N_1$  norm of the density and  $u$  velocity are plotted in figure 4.4a for structured grids and 4.4b for unstructured ones. The CFL imposed are 1 for the RK2 scheme and 30 for the CN scheme. The curves show that the expected second order accuracy is well recovered, confirming the scalar study. Regarding time computations, the resolutions

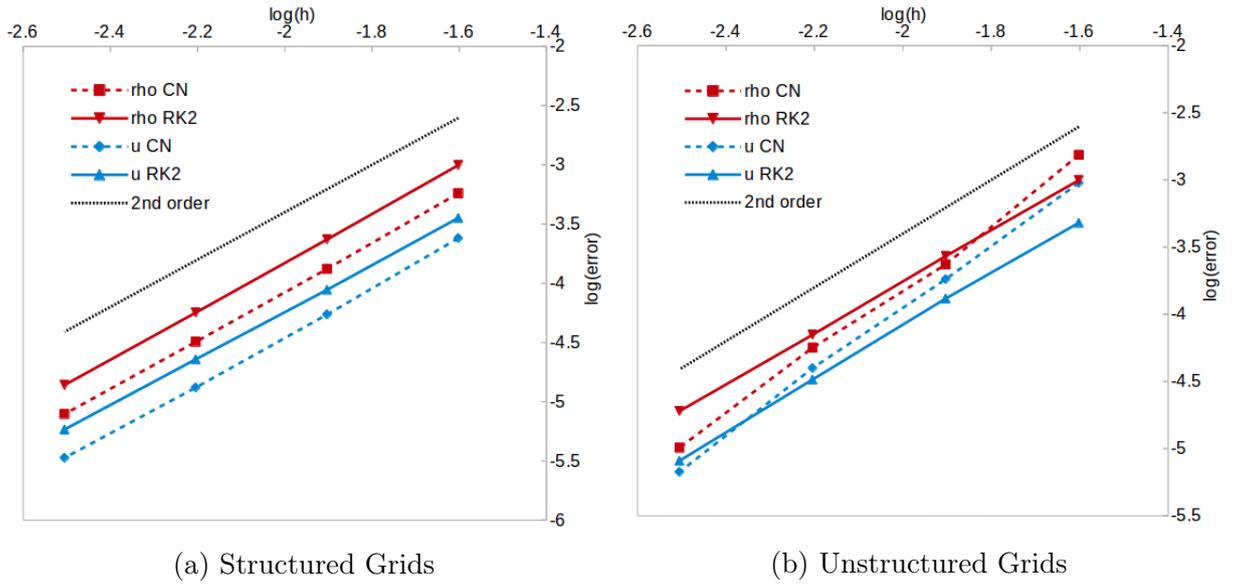


Figure 4.4 – Navier Stokes - ALE Convergence study

using the Cranck Nicolson time discretization are approximatively 12 time faster (for the last two meshes, for coarser ones, the convergence of the Newton procedure requires more iterations). As previously, for a CFL 30 times higher, the computations are only 12 times faster, which is not optimal. However, as already mentioned for non ALE studies, higher CFL are employed for the application test cases leading to a significant gain in term of computational time w.r.t. explicit simulations.

## 4.2 *r*-Adaptation for moving bodies

### 4.2.1 State of the art

The first methodology that can be referred to is the one that is mainly used in the mesh moving techniques that have been proposed in the introduction 1.2.2.1 that is an elasticity based model. In such approaches, the mesh is associated to a material and the classical elasticity equations are solved. We recall here the works mentioned in the introduction of the manuscript, from Tezduyar and co authors [95, 96, 151, 97] and Alauzet and co authors [16, 15]. Another approach that will be denoted "Laplacian based" considers a simpler problem that writes :

$$\nabla_{\mathbf{x}} \cdot (\omega \nabla_{\mathbf{x}} \mathbf{x}) = 0 \quad (4.43)$$

where  $\mathbf{x}$  denotes the position of the nodes and  $\omega$  is called a *monitor function*. In the context of moving bodies, this approach has been used in the already mentioned work of Khurram and Masud [99] instead of the elasticity model. In 2009, Budd *et. al.* proposed a complete article [36] regrouping all the theoretical background. Tang and co authors performed simulations for solving Euler problems with shocks using this technique on quadrangles [156] and triangular meshes [43]. In these references, a conservative interpolation is employed to update the solution from one mesh to another. The same approach has been employed by Ni *et. al.*, but with an ALE resolution, avoiding interpolation procedure [121] (in this last reference, the new position of the nodes is not determined by equation (4.43) but is still dependant on a monitor function). In these works, the monitor function depends on the solution and it takes the form :

$$\omega = \sqrt{1 + \alpha f + \beta g} \quad (4.44)$$

where  $\alpha$  and  $\beta$  are non negative parameter and  $f$  and  $g$  some variables, a common choice being the gradient, and recently the Hessian to avoid too stretched elements [19]. A work quite interesting for our purposes is proposed by Wang *et. al.* [162] for dendritic growth, that requires adaptation to an interface. They modify the monitor function by taking into account a new variable solution of a diffusion equation applied to the heaviside function defining the solid. When solving problem (4.43), a finite element method is employed and as the monitor function needs to be evaluated on the resulting mesh, Newton iterations are performed, but not until convergence, just a limited number of times.

Two approaches have been employed and compared in this work. The first one is to combine the elasticity based approach with the Laplacian based approach. A force dependant on a monitor function is applied on the mesh considered as an elastic material. The second one is to solve directly the Laplacian based model (4.43) with appropriate monitor function. Both of the method are solved using a finite element discretization, and a limited number

of Newton Gauss Seidel (for Elasticity) and Newton Jacobi (for Laplacian) iterations are used to update nodes positions. We first present the methods and the monitor function definitions and then give the advantages and drawbacks for both of them.

## 4.2.2 An Elasticity Based Model

### 4.2.2.1 Problem Statement

The principle of this mesh adaptation approach is to consider the mesh as a material that will be deformed according to the classical Elasticity law :

$$\nabla_{\mathbf{x}} \cdot \underline{\boldsymbol{\sigma}}(\underline{\boldsymbol{\epsilon}}(\mathbf{u})) = \mathbf{F} \quad (4.45)$$

where  $\mathbf{u} = (u_x, u_y)^T$  is the vector of displacement ( $u_x = (x - \chi)$ ,  $u_y = (y - \xi)$ ),  $\underline{\boldsymbol{\sigma}}$  and  $\underline{\boldsymbol{\epsilon}}$  are respectively the constraint and deformation tensors defined as :

$$\underline{\boldsymbol{\epsilon}}(\mathbf{u}) = \frac{1}{2} \left( \nabla_{\mathbf{x}} \mathbf{u} + (\nabla_{\mathbf{x}} \mathbf{u})^T \right) = \begin{pmatrix} \frac{\partial u_x}{\partial \chi} & \frac{\partial u_y}{\partial \xi} \\ \frac{\partial u_x}{\partial \xi} & \frac{\partial u_y}{\partial \chi} \end{pmatrix} \quad (4.46)$$

$$\underline{\boldsymbol{\sigma}}(\underline{\boldsymbol{\epsilon}}(\mathbf{u})) = \begin{pmatrix} (\lambda + 2\mu) \frac{\partial u_x}{\partial \chi} + \lambda \frac{\partial u_y}{\partial \xi} & 2\mu \frac{\partial u_y}{\partial \chi} \\ 2\mu \frac{\partial u_x}{\partial \xi} & (\lambda + 2\mu) \frac{\partial u_y}{\partial \xi} + \lambda \frac{\partial u_x}{\partial \chi} \end{pmatrix} \quad (4.47)$$

with  $\lambda$  and  $\mu$  the Lamé coefficients.

$\mathbf{F}$  is the force applied on the mesh, that determines where the refinement occurs. Thus, the aim is to properly define  $\mathbf{F}$  so as to perform the refinement close to the solid boundaries and, if wanted, to the physics of the problem. As explained previously, it has been chosen to use a monitor function  $\omega$ , based on the idea proposed in [156, 43, 19] to define  $\mathbf{F}$  as :

$$\mathbf{F} = \nabla_{\mathbf{x}} \omega(\mathbf{x}) \quad (4.48)$$

The definition of the monitor function is discussed section 4.2.2.3.

### 4.2.2.2 Finite Element Resolution

Problem (4.45) is rewritten properly with the corresponding BC :

$$\begin{cases} \nabla_{\mathbf{x}} \cdot \underline{\boldsymbol{\sigma}}(\underline{\boldsymbol{\epsilon}}(\mathbf{u})) = \mathbf{F}, & \text{on } \Omega_{\mathbf{x}} \\ \mathbf{u} = \mathbf{0}, & \text{on } \partial\Omega_{\mathbf{x}} \end{cases} \quad (4.49)$$

The BC are chosen such that the boundary of the computational domain is kept fix all over the computation (Dirichlet BC). The weak formulation of this

problem (4.49) writes, considering a test function  $\mathbf{w} = (w_x, w_y) \in [H_0^1]^2$ , find  $\mathbf{u} \in [H_0^1]^2$  such that :

$$\int_{\Omega_{\mathcal{X}}} \underline{\boldsymbol{\epsilon}}(\mathbf{w}) : \underline{\boldsymbol{\sigma}}(\mathbf{u}) = \int_{\Omega_{\mathcal{X}}} \mathbf{F} \cdot \mathbf{w}, \quad \forall \mathbf{w} \in [H_0^1]^2 \quad (4.50)$$

where  $\underline{A} : \underline{B} = \sum_i \sum_j A_{ij} B_{ji}$  is the double dot product.

Before discretizing this equation, let rewrite it over a more practical way :

$$\begin{aligned} \int_{\Omega_{\mathcal{X}}} ([\underline{\mathcal{M}}_1 \nabla_{\mathcal{X}} u_x + \underline{\mathcal{M}}_2 \nabla_{\mathcal{X}} u_y] \cdot \nabla_{\mathcal{X}} w_x + [\underline{\mathcal{M}}_3 \nabla_{\mathcal{X}} u_x + \underline{\mathcal{M}}_4 \nabla_{\mathcal{X}} u_y] \cdot \nabla_{\mathcal{X}} w_y) &= \int_{\Omega_{\mathcal{X}}} \mathbf{F} \cdot \mathbf{w} \\ \underline{\mathcal{M}}_1 &= \begin{pmatrix} \lambda + 2\mu & 0 \\ 0 & \mu \end{pmatrix}, \underline{\mathcal{M}}_2 = \begin{pmatrix} 0 & \lambda \\ \mu & 0 \end{pmatrix}, \underline{\mathcal{M}}_3 = \begin{pmatrix} 0 & \mu \\ \lambda & 0 \end{pmatrix}, \underline{\mathcal{M}}_4 = \begin{pmatrix} \mu & 0 \\ 0 & \lambda + 2\mu \end{pmatrix} \end{aligned} \quad (4.51)$$

The discretization of this problem (4.51) is done using a  $P_1$  continuous Galerkin FE method. A triangulation  $\mathcal{T}^h$  of the domain is considered on which the approximated displacement are defined using the Lagrange basis function  $u_{x/y} = \sum_{i \in \mathcal{T}^h} u_{x/y_i} \varphi_i$ . Considering the two functions  $\mathbf{w} = (\varphi_i, 0)$  and  $\mathbf{w} = (0, \varphi_i)$ , the scheme writes :

$$\forall i, \begin{cases} \sum_{T \ni i} \left( \sum_{j \in T} u_{x_j} \int_T [\underline{\mathcal{M}}_1 \nabla \varphi_j] \cdot \varphi_i + \sum_{j \in T} u_{y_j} \int_T [\underline{\mathcal{M}}_2 \nabla \varphi_j] \cdot \varphi_i \right) = \sum_{T \ni i} \sum_{j \in T} F_{x_j} \int_T \varphi_j \varphi_i \\ \sum_{T \ni i} \left( \sum_{j \in T} u_{x_j} \int_T [\underline{\mathcal{M}}_3 \nabla \varphi_j] \cdot \varphi_i + \sum_{j \in T} u_{y_j} \int_T [\underline{\mathcal{M}}_4 \nabla \varphi_j] \cdot \varphi_i \right) = \sum_{T \ni i} \sum_{j \in T} F_{y_j} \int_T \varphi_j \varphi_i \end{cases}$$

This system (4.2.2.2) can be written over the following matrix form :

$$\begin{aligned} \underline{\mathbf{K}} \mathbf{U} &= \underline{\mathbf{M}} \mathbf{F} \\ \underline{\mathbf{K}} &= \begin{pmatrix} \underline{\mathbf{K}}_1 & \underline{\mathbf{K}}_2 \\ \underline{\mathbf{K}}_3 & \underline{\mathbf{K}}_4 \end{pmatrix}, \mathbf{U} = \begin{pmatrix} \mathbf{U}_x \\ \mathbf{U}_y \end{pmatrix}, \underline{\mathbf{M}} = \begin{pmatrix} \underline{\mathbf{M}}^{Gal} & 0 \\ 0 & \underline{\mathbf{M}}^{Gal} \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \mathbf{F}_x \\ \mathbf{F}_y \end{pmatrix} \end{aligned} \quad (4.52)$$

where  $\mathbf{U}_x = (u_{x_1}, \dots, u_{x_{N_v}})^T$  and  $\mathbf{U}_y = (u_{y_1}, \dots, u_{y_{N_v}})^T$  are the displacement vectors of the  $x$  and  $y$  component of the nodes.  $\underline{\mathbf{M}}^{Gal}$  is the Galerkin mass matrix of size  $N_v \times N_v$  that writes for second order Lagrange basis function (two dimensional case) :

$$M_{ij}^{Gal} = \sum_{T \ni i} \sum_{j \in T} \varphi_i \varphi_j = \begin{cases} \sum_{T \ni i} \frac{|T|}{6}, & \text{if } i = j \\ \sum_{T \ni i} \frac{|T|}{12}, & \text{if } i \neq j \end{cases} \quad (4.53)$$

The  $\underline{\mathcal{K}}_{\alpha, \alpha=1,2,3,4}$  are defined by (two dimensional case) :

$$\mathcal{K}_{\alpha_{ij}} = \sum_{T \ni i} \sum_{j \in T} \frac{1}{4|T|} [\mathcal{M}_{\alpha} \mathbf{n}_j] \cdot \mathbf{n}_i \quad (4.54)$$

The matrices  $\underline{\mathbf{M}}^{Gal}$  and  $\underline{\mathcal{K}}_{\alpha}$  are defined on the reference mesh : they only need to be evaluated at the beginning of the computation. In the RHS, the forces are evaluated on the current mesh. Indeed, coming back on its definition (4.48) :

$$F_x = \frac{\partial \omega(\mathbf{x})}{\partial \chi}, \quad F_y = \frac{\partial \omega(\mathbf{x})}{\partial \xi}$$

where  $\mathbf{x}$  denotes the nodal coordinates of the adapted mesh. Thus, a limited number of Newton Gauss-Seidel process is performed at each time step, with as initial condition the position of the mesh at previous time step. During this procedure, at each Newton iteration, a transitional mesh is defined, on which the relevant values are interpolated from the initial mesh, to compute the new monitor function (the signed distance function and physical variable of adaptation).

**Remark 4.4.** The number of iterations is dependant on the time step. The idea proposed in [156, 43, 19] is to start from the mesh at time  $t^n$  that is assumed close to the converged solution of the mesh adaptation problem at time  $t^{n+1}$ . Thus, the number of iterations necessary to achieve the same degree of convergence increases with the time step.

#### 4.2.2.3 Monitor function definition

As we are willing to refine the mesh close to the interface and according to a physical parameter, the monitor function has to be defined according to both of those criteria. Thus, two monitor functions are defined, one for the level set  $\psi$  denoted  $\omega_{\psi}$  and one for the chosen physical variable  $v$  denoted  $\omega_v$ . For the monitor function associated to the level set adaptation, it has been chosen to use :

$$\omega_{\psi} = \sqrt{\alpha_{\psi} e^{-\beta_{\psi} \psi^2}} \quad (4.55)$$

The basic idea is to impose a force close the boundary of the solid and the elastic property of the "material" perform a kind of gradation avoiding to have to large element as soon as this force is no more applied. Figure 4.5 is an example of adaptation for a circle ( $r = 0.5$ ) embedded in a squared mesh of sizes  $[-2, 2]^2$  of 3927 vertices with  $\alpha_{LS} = 100$ ,  $\beta_{\psi} = 40$ . In some cases, this monitor function can in addition take into account the curvature of the level set (denoted by  $\kappa$ ) in the vicinity of the interface :

$$\omega_{\psi} = \sqrt{\alpha_{\psi} e^{-\beta_{\psi} \psi^2} + \alpha_{\kappa} |\kappa|} \quad (4.56)$$

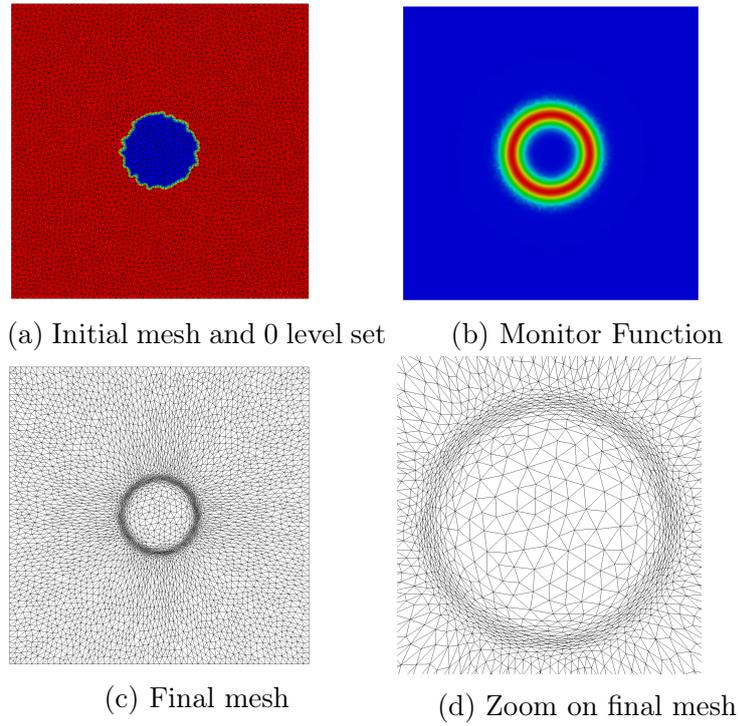


Figure 4.5 – Elasticity level set adaptation

For the physical adaptation, the monitor function, as proposed in [156, 43] is based on the gradient of the solution  $v$ . However, as we deal with relatively smooth solutions, the gradient is "truncated" to better take into account smoother features. Thus, the "truncated" gradient  $\widetilde{\nabla}v$  is defined as :

$$\widetilde{\nabla}v = \min(\beta_v \nabla v, \sup \nabla v) \quad (4.57)$$

and the monitor function  $\omega_v$  is defined by :

$$\omega_v = \sqrt{\alpha_v \|\widetilde{\nabla}v\|} \quad (4.58)$$

Once both of those monitor functions are defined, they have to be genuinely combined to perform a proper adaptation (one being not preponderant with respect to the other). Thus,  $\omega_v$  is modified so as to ensure to have  $\omega_\psi$  and  $\omega_v$  of the same order of magnitude. Indeed, as  $\sup e^{-\alpha_\psi \psi^2} = 1$ , by using the normalized norm of the gradient  $\|\nabla v\|_n$  in (4.58), the monitor functions are only pondered by  $\alpha_\psi$  and  $\alpha_v$  :

$$\omega_v = \sqrt{\alpha_v \|\widetilde{\nabla}v\|_n} \quad (4.59)$$

In addition, so as to ensure to have the wanted adaptation close to the interface (which is mandatory to improve the interface definition), around an area  $\epsilon$  of

the 0 level set, only the monitor function associated to the surface is considered. Thus, the monitor function, combination of (4.59,4.55) can be defined by :

$$\omega = \begin{cases} \omega_\psi, & \text{if } \psi < \epsilon \\ \max(\omega_\psi, \omega_v), & \text{if } \psi \geq \epsilon \end{cases} \quad (4.60)$$

To illustrate this definition, a profile of  $u$  velocity on a referential mesh (10, 647 vertices) is proposed figure figure 4.6a (coming from the oscillating cylinder case fully described in the next section 4.3.2.1). The monitor function computed with  $\alpha_\psi = \alpha_v = 40$ ,  $\beta_\psi = 100$  and  $\beta_v = 2.5$  are given figure 4.6e and 4.6f. To emphasize the importance of using a "truncated" gradient, the same adaptation is performed with  $\beta_v = 1$ . The resulting monitor function and corresponding mesh are plotted figure 4.6c and 4.6d. We can see that the mesh adapted using the truncated gradient proposes a better refinement w.r.t. the solution, without degrading the interface adaptation. Thus, the resolution of the flow should be improved in the simulations by performing the truncation of the gradient in an adequate manner.

## 4.2.3 A Laplacian Based Model

### 4.2.3.1 Problem Statement

As explained in the introduction of this section, the problem (4.43,4.44) can be solved. In this expression (4.43),  $\omega$  is originally a matrix, but as in the works already mentioned [156, 43, 19], the dimensions are uncoupled by choosing the identity matrix multiplied by the monitor function. The mesh PDE problem now reads :

$$\nabla_{\chi} \cdot (\omega \nabla_{\chi} x) = 0 \quad (4.61)$$

which is applied independently on all of the directions. The coupling arises from the definition of the smoothness sensor  $\omega$ . This method can be basically seen as an equidistribution of the nodes according to the monitor function. Indeed, (4.61) can be rewritten as :

$$\omega \nabla_{\chi} x = C^t \quad (4.62)$$

Thus, for high values of  $\omega$ , the nodes are concentrated, and on the opposite, for low values, the nodes are moved aside. The monitor function does not play exactly the same role than for the Elasticity based model. Here, it equidistributes the nodes whereas for the previous model, it was defining the forces deforming the mesh. Problem (4.61) can be rewritten in terms of displacement as :

$$\nabla_{\chi} \cdot (\omega \nabla_{\chi} u) = -\nabla_{\chi} \cdot (\omega \nabla_{\chi} x_0) \quad (4.63)$$

with  $x = x_0 + u$ , and  $x_0$  the initial position, which is not necessary the referential position  $\chi$ . For instance, in the simulations,  $x_0$  is the position of the nodes at the previous time steps.

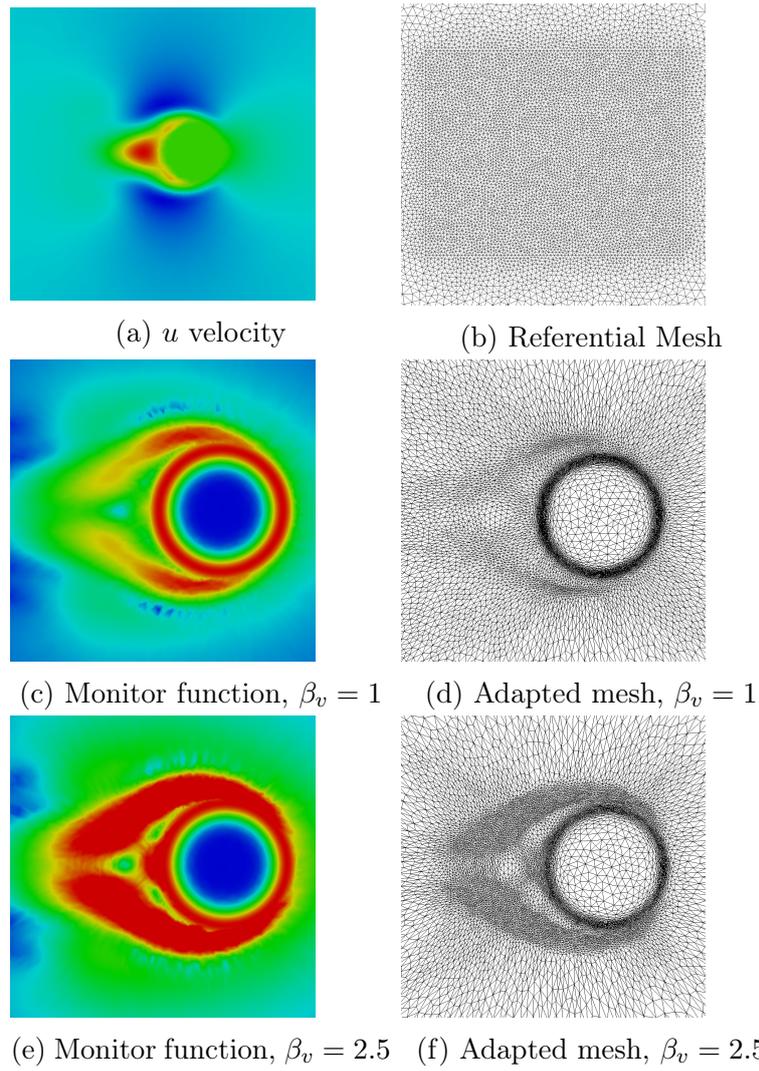


Figure 4.6 – Combination of physical and level set adaptation

### 4.2.3.2 Finite Element Resolution

We rewrite the problem expressed on the displacement with proper BC :

$$\begin{cases} \nabla_{\mathbf{x}} \cdot (\omega \nabla_{\mathbf{x}} u) = -\nabla_{\mathbf{x}} \cdot (\omega \nabla_{\mathbf{x}} x_0) & \text{on } \Omega_{\mathbf{x}} \\ u = 0, & \text{on } \partial\Omega_{\mathbf{x}} \end{cases} \quad (4.64)$$

As for the elasticity based model 4.2.2, homogeneous Dirichlet BC are imposed to keep the nodes of the boundaries of the computational domain fix. Using these hypotheses, the classical variational form of the problem in  $H_1^0$  reads :

$$\int_{\Omega_{\mathbf{x}}} \omega \nabla_{\mathbf{x}} u \cdot \nabla_{\mathbf{x}} w = - \int_{\Omega_{\mathbf{x}}} \omega \nabla_{\mathbf{x}} x_0 \cdot \nabla_{\mathbf{x}} w \quad (4.65)$$

The discretization is still performed using a  $P_1$  continuous Galerkin FE method. A mesh  $\mathcal{T}^h$  of the domain is considered and the approximated displacement and initial positions writes  $u_h = \sum_{j \in \mathcal{T}^h} u_j \varphi_j$  and  $x_{0_h} = \sum_{j \in \mathcal{T}^h} x_{0_j} \varphi_j$ . Considering the function  $w = \varphi_i$ , the scheme writes :

$$\forall i, \sum_{T \ni i} \sum_{j \in T} u_j \int_T \omega \nabla \varphi_i \cdot \nabla \varphi_j = - \underbrace{\sum_{T \ni i} \sum_{j \in T} x_{0_j} \int_T \omega \nabla \varphi_i \cdot \nabla \varphi_j}_{b_i} \quad (4.66)$$

that can be written over a matrix form :

$$\underline{\mathbf{K}} \mathbf{U} = \mathbf{B} \quad (4.67)$$

with  $\mathbf{U}$  is the vector of displacement of size  $N_v$ ,  $\mathbf{B} = (b_1, \dots, b_{N_v})^T$  and  $\underline{\mathbf{K}}$  is of size  $N_v \times N_v$  defines by :

$$K_{ij} = \sum_{T \ni i} \sum_{j \in T} \int_T \omega \nabla \varphi_i \cdot \nabla \varphi_j \quad (4.68)$$

As we consider  $P_1$  approximations, denoting by  $\bar{\omega}^T = \sum_{j \in T} \frac{\omega_j}{3}$  the mean value of  $\omega$  on the element, and using (2.44), we have :

$$K_{ij} = \sum_{T \ni i} \sum_{j \in T} \bar{\omega} \frac{\mathbf{n}_i \cdot \mathbf{n}_j}{4|T|} \quad (4.69)$$

As proposed in [156, 43, 19] and for the elasticity model,  $\omega$  depends on the current position of the node  $\mathbf{x}$ , requiring the use of a newton procedure, to evaluate its new values at each iterations. As explained for the Elasticity based model, at each iteration of the Newton procedure, the relevant values are interpolated from the initial mesh to the temporary one to evaluate the new monitor function.

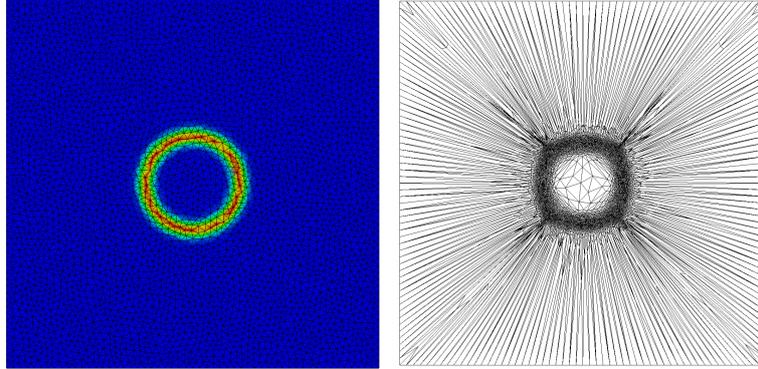


Figure 4.7 – Laplacian Adaptation - Null monitor function. Left : Initial mesh and monitor function (blue :  $3.10^{-4}$ , red : 6.32) - Right : Resulting Mesh

### 4.2.3.3 Monitor Function Definition

#### Classical definition

As explained previously, the monitor function has here a different role w.r.t. what has been done for the Elasticity based approach. Thus, it has to be defined differently. Indeed, as recalled in the state of the art 4.2.1, the monitor function is often defined as :

$$\omega = \sqrt{1 + \alpha f + \beta g} \quad (4.70)$$

with  $f$  and  $g$  the gradient and hessian of the physical variable chosen for the adaptation. Looking at equation (4.62), we see that if  $\omega \rightarrow 0$ ,  $\nabla x \rightarrow \infty$ . Thus the monitor function must be chosen non null in all nodes of the mesh. The definition of section 4.2.2.3 cannot be used. For instance, considering only an adaptation to the 0 level set of the solid, applying the monitor function (4.55) concentrates all the points of the mesh in the vicinity of the interface, as illustrated figure 4.7, with  $\alpha_\psi = 40$  and  $\beta_\psi = 400$  for the circle of radius 0.5 in a squared mesh of sizes  $[-2, 2]^2$  of 3927 vertices.

Thus, the monitor function must be defined over a form (4.70), or a similar one such as :

$$\omega = \sqrt{1 + \alpha_\psi e^{-\beta_\psi \psi^2}} \quad (4.71)$$

and to adapt to a physical variable, we define as previously the truncated gradient (4.57) and compute the monitor function as introduced in the literature :

$$\omega_v = \sqrt{1 + \alpha_v \|\widetilde{\nabla v}\|_n} \quad (4.72)$$

For instance, the same circle adaptation is performed with (4.71), with the same parameters :  $\alpha_\psi = 40$  and  $\beta_\psi = 400$  and the resulted mesh is well adapted to the 0 level set (see figure 4.8).

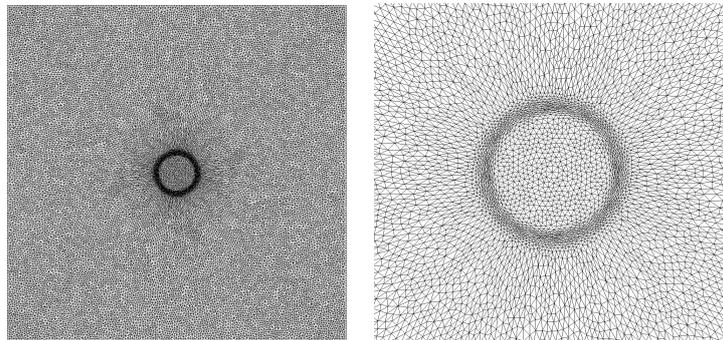


Figure 4.8 – Circle Adaptation, Laplacian Based 'classical' approach. Left : General View - Right : Zoom

### Constant by part definition

Another way of defining the monitor function has also been studied for this method, inspired from the work of Farhat and Lakshminarayan [68]. In the last reference, the authors suggest to define an area around the body that is translated and rotated according to the movement of the boundary. The application considered in the reference is however only a limited displacement of bodies (e.g. flapping wings). Based on this idea, we propose to define a monitor function that will allow to keep all along the simulation a refined area around the interface, so as to provide an accurate resolution of the physics. In this refined area, an adaptation to the interface is added so as to define in a best possible way the boundary. In comparison with the approach of Farhat and Lakshminarayan, this adaptation allows to perform larger displacements, such as the study of ice shedding trajectories. The idea is very simple and is based on a monitor function defined constant by part according to the signed distance function :

$$\omega_\psi = \begin{cases} C_1, & \text{if } |\psi| < w_1 \\ C_2, & \text{if } w_1 \leq |\psi| < w_2 \\ C_3, & \text{elsewhere} \end{cases} \quad (4.73)$$

with  $C_1 > C_2 > C_3$  constants. For instance, considering the same illustration than previously, the circle is adapted with 4 layers :  $w_1 = 0.05$ ,  $w_2 = 1$ ,  $w_3 = 1.75$  and  $C_1 = 225$ ,  $C_2 = 90$ ,  $C_3 = 70$ ,  $C_4 = 20$  and the resulted mesh is displayed figure 4.9. Such a procedure allows to use more nodes far from the interface to propose a refined area in a larger vicinity of the 0 level set according to the chosen adaptation parameters.

As for the elasticity based approach, the SDF and physical adaptation can be coupled to define a monitor function over the form (4.60). The truncated gradient needs however to be normalized according to the maximum value of

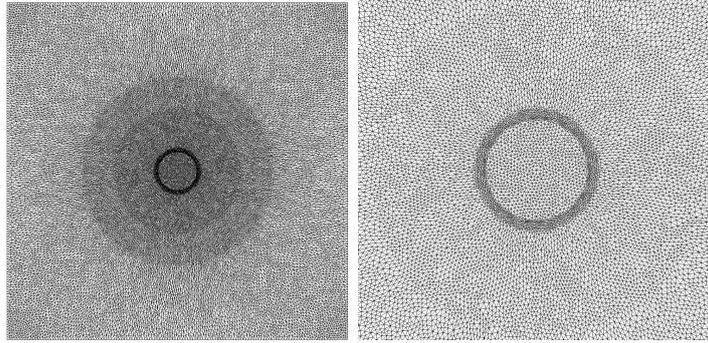


Figure 4.9 – Circle Adaptation, Laplacian Based constant by part approach. Left : General View - Right : Zoom

the  $\omega_\psi$  definition :

$$\|\widetilde{\nabla v}\|_n = \frac{\|\widetilde{\nabla v}\|}{\sup \|\widetilde{\nabla v}\|} \sup \omega_\psi \quad (4.74)$$

where in the case of the constant by part definition,  $\sup \omega_\psi = C_1$ .

Now, we study the advantages and drawbacks of the different proposed approaches. The adaptation to a complex geometry is proposed in the next section to explain what are the choices made in the applications that are proposed in the section of results 4.3.2.

#### 4.2.4 Comparison of the approaches

Two methods have been presented in the two previous sections. The question is to try to evaluate which one present the best properties. This is a tricky point, and the choice will finally be dependant of the test case studied.

The first noticeable difference is the matrix computation. Indeed, in the Elasticity based model 4.2.2, the matrix is defined on the referential mesh. It means that it only need to be evaluated at the beginning of the computation, and is employed during the whole simulation. The only element that gives the non linearity of the problem, meaning the dependency on the monitor function, is the right hand side. On the contrary, for the Laplacian based adaptation 4.2.3, the matrix itself is dependant on the monitor function. Thus, it requires to be evaluated at each Newton iterations of each time step. The use of the elasticity model is thus strongly wanted for sake of computational time. In addition, the Elasticity model coupling the displacement in the different space dimensions, a natural anisotropy is performed, that is more difficult to obtain with the simpler model that is the Laplacian based approach.

However, for complex geometries, more precisely non convex ones, this Elasticity method suffers from the difficulty to handle singularities, that the

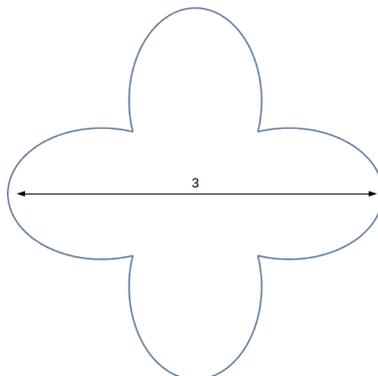


Figure 4.10 – Flower surface for adaptation

Laplacian based approach overcomes in a very natural way. For instance, let consider the 'flower' displayed figure 4.10 embedded in a square of sizes  $[-4, 4]^2$  (36237 vertices). Figures 4.12a, 4.12b and 4.12c proposes the adaptation with Elasticity approach,  $\alpha_\psi = 30$ ,  $\beta_\psi = 200$  and  $\alpha_\kappa = 2.5$  (figure 4.12a), the Laplacian based approach with "classical" definition of the monitor function,  $\alpha_\psi = 40$ ,  $\beta_\psi = 350$  (figure 4.12b) and the Laplacian based approach with the monitor function defined constant by part,  $C_1 = 315$ ,  $C_2 = 150$ ,  $C_3 = 75$ ,  $C_4 = 10$  and  $w_1 = 0.05$ ,  $w_2 = 0.75$ ,  $w_3 = 1.75$  (figure 4.12c). Those adaptations illustrate the previous points : the elasticity proposes a nice natural anisotropy, but the Laplacian approach handle a lot better discontinuities. In addition, we compare the mesh close to the boundary for the two monitor function definitions in the Laplacian approach figure 4.11. The idea of the monitor function constant by part allows to have a mesh that is more suitable to perform computations as from the same mesh we propose a better refinement close the interface.

That is why, when 'simple' geometries are involved, the elasticity based approach is employed, because it handles nicely the interface adaptation, and reduces the computational cost as the matrix is only evaluated at the beginning of the simulations. For more complex ones (such as ice shedding simulations), the Laplacian model is employed with a definition of the monitor function constant by part because of the better handling of the singularities, and a suitable mesh to solve physics in a vicinity of the boundary, even without adaptation to a physical variable.

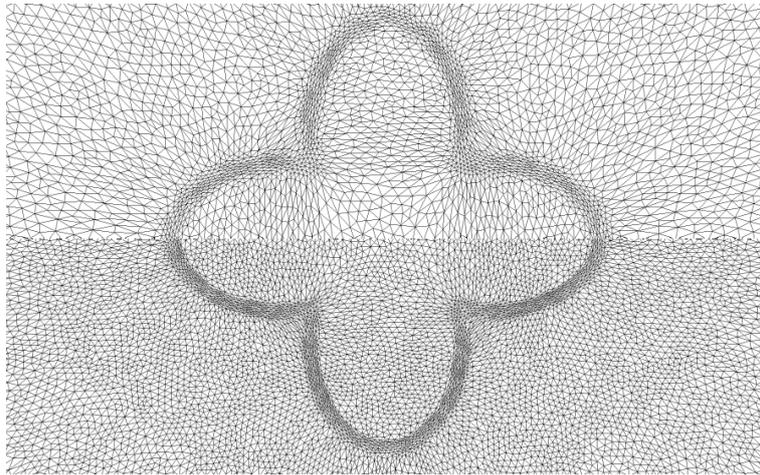
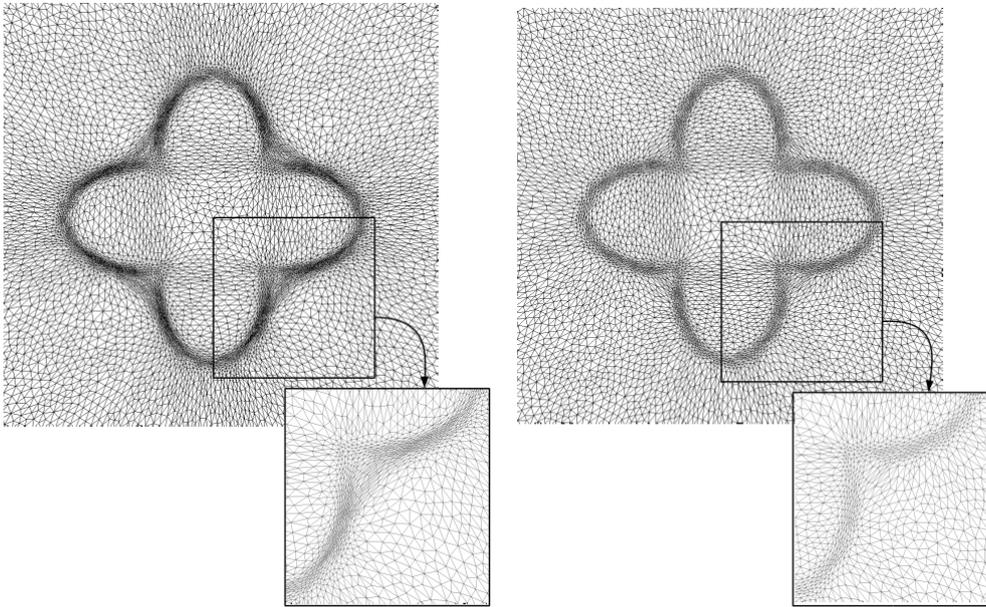
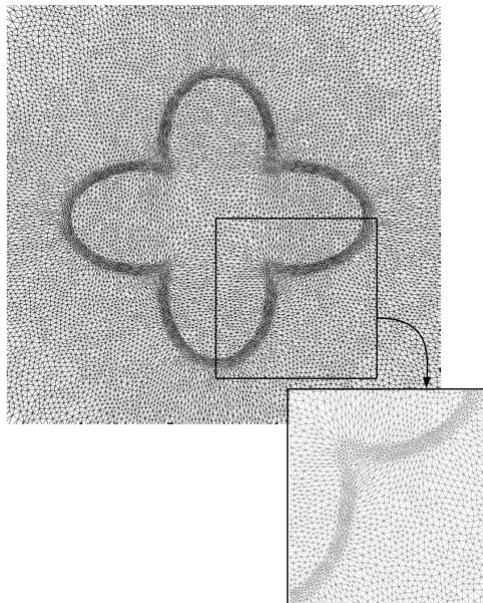


Figure 4.11 – Flower Adaptation, Comparison of monitor function definition.  
Top : 'Classical' - Bottom : 'Constant by part'



(a) Flower Adaptation - Elasticity Approach (b) Flower Adaptation - Laplacian Approach, 'classical' definition



(c) Flower Adaptation - Laplacian Approach, constant by part definition

Figure 4.12 – Flower Adaptation Comparison

## 4.3 Penalized Navier Stokes Equations with Moving Bodies

To perform simulations of moving bodies with penalized Navier Stokes equations, the previous proposed r-adaptivity are coupled to the ALE resolution of the equations. After an explanation of the sequence to couple the different tools presented, several 2D test cases are proposed in order to validate the proposed approach. The two first ones are characterized by a solid velocity imposed by an analytical function. The last one is a simulation in which full FSI is simulated. The solid velocity is ruled by external forces that are the gravity and the aerodynamical forces. Finally, this method is applied to the simulation of a GLC305 rice ice shape, focus of the STORM project.

### 4.3.1 Moving object simulation - process

All the tools necessary to handle moving bodies on adaptive unstructured grids using an IB method are now settled and can be gathered so as to perform the wanted simulations. We propose here the different steps to perform such studies. Indeed, as we consider moving bodies, different stages have to be arranged at each time of the resolution (level set advection, mesh adaptation, and splitting steps). We first explain the strategy adopted to perform the displacement of the body before giving the algorithm for the process iterated at each time step.

#### Solid Motion

The solid is known implicitly *via* the 0 level set of the signed distance function. Thus, when considering moving bodies, the signed distance function  $\psi$  has to be advected according to the solid velocity :

$$\frac{\partial \psi}{\partial t} + \mathbf{u}_s \cdot \nabla \psi = 0 \quad (4.75)$$

A well known problem when advecting a level set function is the non conservation, characterized by a gain or loss of mass that involves the need of redistanciation (see among others [37, 157, 155]) with high order discretization schemes such as 4<sup>th</sup> order Runge Kutta [37]. However, in our simulations, a surfacic mesh of our solid can be provided (so as to compute the initial signed distance function).

In a first time, as only 2D simulations are performed, from a computational point of view, it is simpler and more efficient to advect directly this mesh and to recompute the signed distance function than solving (4.75), performing redistanciation when necessary.

### FSI simulation - Process

To describe the algorithm employed, the following notations are defined : the solid at time  $t^n$  is defined by its level set function  $\psi^n$  on the adapted mesh  $\mathcal{T}_h^n$  (coordinates denoted by  $\mathbf{x}^n$ ). The level set function is associated to the surfacic mesh of the solid whom coordinates are denoted  $\mathbf{x}_s^n$ . The forces exerted on the solid are denoted  $\mathbf{F}^n$ , the solid velocity is denoted  $\mathbf{u}_s^n$  and its angular velocity  $\boldsymbol{\theta}^n$ . The different steps are the following :

- *Step 1 : Solid Velocity update.* If the solid is moved by an analytical function  $\mathbf{f}(\mathbf{x}, t)$ , as proposed in the two first test cases (section 4.3.2.1 and 4.3.2.2), then  $\mathbf{u}_s^{n+1} = \mathbf{f}(\mathbf{x}, t^{n+1})$ . If the motion is ruled by aerodynamical forces  $\mathbf{F}^n$  and external forces  $\mathbf{F}^{ext}$ , as proposed in the third test case (section 4.3.2.3), then, as proposed in [27], we solve

$$\begin{aligned} m_a \frac{\partial \mathbf{u}_s}{\partial t} &= \mathbf{F}^n + \mathbf{F}^{ext} \\ m_i \frac{\partial \boldsymbol{\theta}}{\partial t} &= \mathbf{T} \end{aligned} \quad (4.76)$$

where  $m_a$  is the solid mass and  $m_i$  its moment of inertia. A simple explicit Euler scheme is used (more sophisticated discretization could be employed) :

$$\begin{aligned} \mathbf{u}_s^{n+1} &= \mathbf{u}_s^n + \Delta t \frac{\mathbf{F}^n + \mathbf{F}^{ext}}{m_a} \\ \boldsymbol{\theta}^{n+1} &= \boldsymbol{\theta}^n + \Delta t \frac{\mathbf{T}}{m_i} \end{aligned} \quad (4.77)$$

- *Step 2 : Object Displacement.* Using this velocity  $\mathbf{u}_s^{n+1}$ , employing once again an Euler discretization, the nodes of the surfacic mesh are updated by :

$$\mathbf{x}_s^{n+1} = \mathbf{x}_s^n + \Delta t \mathbf{u}_s^{n+1} \quad (4.78)$$

From this new surfacic mesh is computed the level set function  $\psi^{n+1}$ .

- *Step 3 : Mesh Adaptation.* The mesh is adapted to the new level set defining the solid  $\psi^{n+1}$  and the physical solution  $\mathbf{u}_h^n$ . The mesh velocity necessary for the ALE resolution is computed :  $\boldsymbol{\sigma} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t}$ .
- *Step 4 : Fluid Resolution.* The ALE resolution of the penalized Navier Stokes equations is performed. The solution  $u_h^{n+1}$  at time  $t^{n+1}$  is obtained. Before the last step of the splitting, the computation of the aerodynamical forces with the change of momentum approach (section 3.2.4.1) is performed :  $\mathbf{F}_{CM}^{n+1}$ .

- *Step 5 : Integral Force Computation* : The solution  $\mathbf{u}_h^{n+1}$  is interpolated on the surfacic mesh defined by  $\mathbf{x}_s^{n+1}$  and the aerodynamical forces  $\mathbf{F}_{IC}$  are evaluated by integrating shear stress and pressure contribution (section 3.2.4.2). This force evaluation is the one employed in (4.76) when dealing with motion ruled by external forces.

This process is now validated on several test cases. The oscillating cylinder and naca are performed to assess the validity of the approach and the forces computations for moving solid cases. The falling cylinder validates the full process, when the solid velocity is directly dependent on the aerodynamical forces. Finally, we propose to apply this technique to the study of a 2D ice shape, concerns of the STORM project, without validation as the Reynolds number employed for the simulation is lower than the real one by lack of turbulence model.

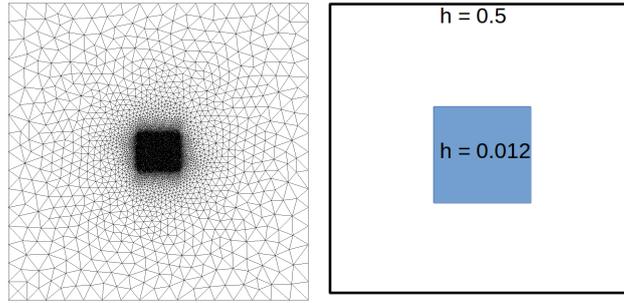


Figure 4.13 – Referential mesh for the oscillating cylinder test case and corresponding sizes

## 4.3.2 Results

### 4.3.2.1 In-line oscillating cylinder in a flow at rest

For this test case, a cylinder ( $D = 0.2$ ) is oscillating in a fluid at rest. The motion is ruled by :

$$x(t) = -A \sin(2\pi ft) \quad (4.79)$$

where  $A$  is the amplitude of the oscillation and  $f$  its frequency. The dimensionless number characterising this case are the Reynolds number  $Re = \frac{U_{max} D}{\nu} = 100$  and the Keulegan-Carpenter number  $KC = \frac{2\pi f}{D}$ . The computational domain is  $[-10, 10] \times [-8, 8]$  and the referential mesh (13424 vertices and 26782 triangles) is given figure 4.13 along with the sizes used to generate this mesh. For the adaptation, the elasticity model is considered and the parameters used for the monitor function (4.60) are  $\alpha_\psi = \alpha_v = 40$ ,  $\beta_\psi = 100$  and  $\beta_v = 2.5$ , and the physical variable used is the  $u$  velocity.

For this test case, the ALE RK2 RD scheme is employed with a  $CFL = 1$ .

The number of Gauss Seidel iteration for the adaptation process is set to 10. Plot of the velocity and the corresponding adapted mesh are presented figure 4.14. We can see that the adaptation process allows to have an optimisation of the mesh close to the 0 level set leading to an accurate definition of the solid, and that the mesh adapts well to the physics of the problem.

In order to validate these results, cuts are done at different times and at different positions and compared to the literature. The comparison is done with experimental data (Dütsch *et al.* [66]), the Lescape computational code (penalization on structure grids, see Beaugendre and Morency [27]), and the IBM proposed by Liao *et al.* [109]. The plots figure 4.15 show that the present simulation is in good agreement with the literature, but with 13424 vertices compared to the 50000 used in [109] and 1200000 used in the Lescape code (the mesh being uniform in the whole domain). It demonstrates the benefit provided by the combination unstructured mesh adaptation/IB as we severely diminish the number of nodes required to perform accurate simulations. In addition, to illustrate the gain of accuracy thanks to the adaptation process, we plot the 0 level set on the adapted mesh and initial one figure 4.16. With the same approach, the simulation would not have been possible on the original mesh without adaptation due to the lack of accuracy in the solid definition.

#### 4.3.2.2 Oscillating Naca Airfoil

Flapping wing motions are extensively studied for engineering applications in low Reynolds numbers flow where classical fixed wing geometry performance decreases, [75]. According to previous works, around ten parameters influence the power extraction in flapping wing motions, such as oscillation frequencies and amplitudes (translational and rotational), phase difference between plunge and pitch motion, viscosity, free stream velocity, flapping pattern and airfoil geometry.

In this section, an oscillating airfoil experiencing simultaneous pitching  $\theta(t)$  and heaving  $h(t)$  motions is modelled. The infinitely long wing is based on a NACA 0015 airfoil. The pitching axis is located along the airfoil chord at the position  $(x_p, y_p) = (1/3, 0)$ . The airfoil motion, described by Kinsey and Dumas [101], is defined by the heaving  $h(t)$  and the pitching angle  $\theta(t)$  as follows

$$\begin{cases} h(t) = H_0 \sin(\omega t + \Phi) \\ \theta(t) = \theta_0 \sin(\omega t) \end{cases} \quad (4.80)$$

where  $H_0$  is the heaving amplitude and  $\theta_0$  is the pitching amplitude. The angular frequency is defined by  $\omega = 2\pi f$  and the phase difference  $\Phi$  is set to  $90^\circ$ . The heaving velocity is then given by

$$V_y(t) = H_0 \omega \cos(\omega t + \Phi). \quad (4.81)$$

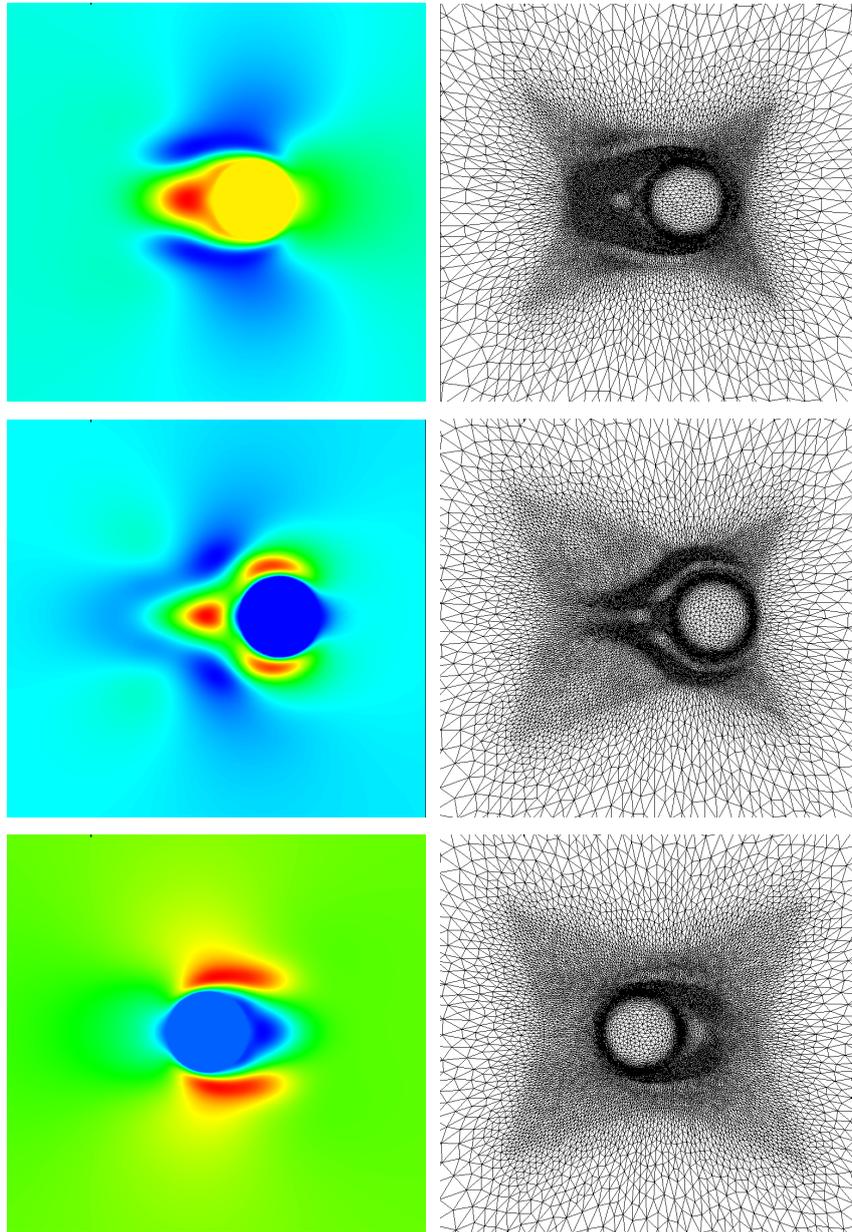
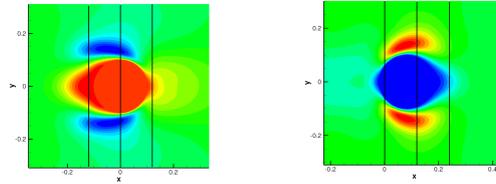
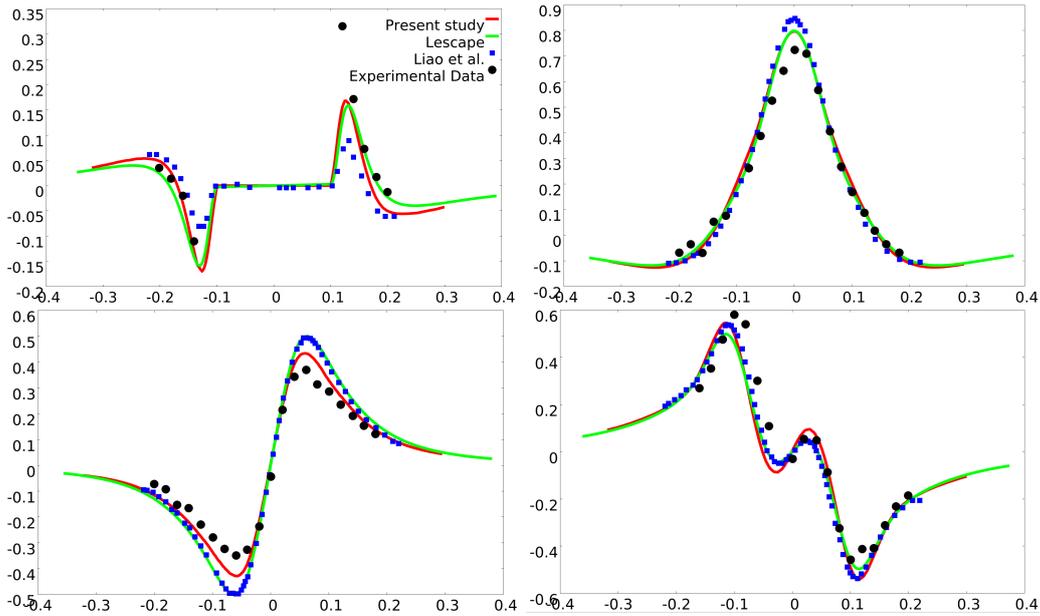


Figure 4.14 –  $u$  velocity and corresponding adapted mesh at different time (zoom close to the adapted area).

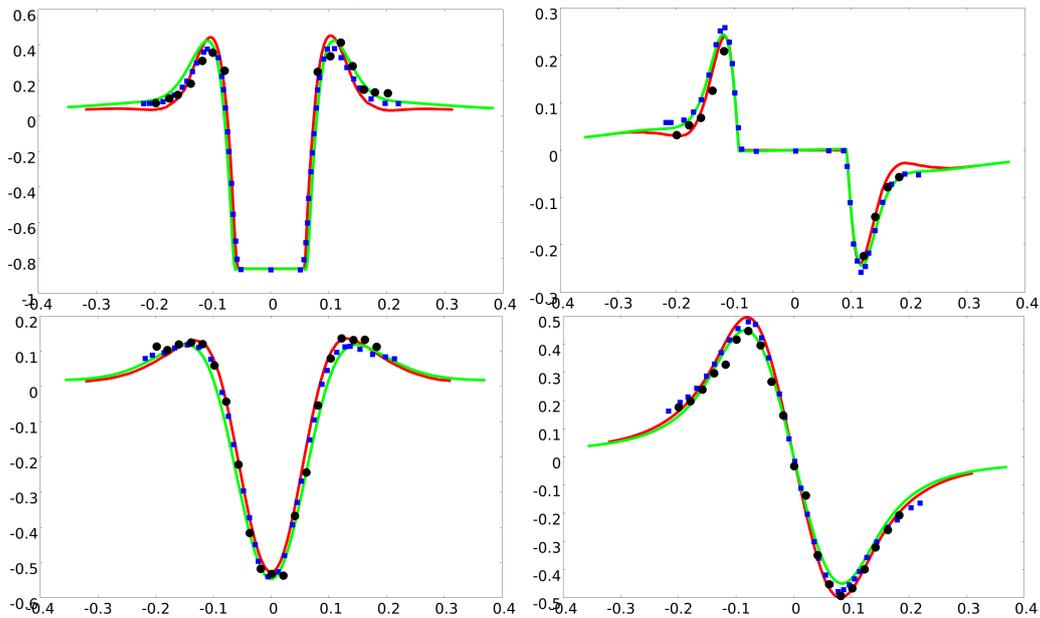
4. Penalization and Moving Bodies - An ALE Approach



(a) Cuts performed at different times



(b) cuts at 180. From top left to bottom right :  $x = 0D$ ,  $v$  velocity -  $x = 6D$ ,  $u$  velocity -  $x = 6D$   $v$  velocity,  $x = -6D$   $v$  velocity



(c) cuts at 330. From top left to bottom right :  $x = 0D$ ,  $u$  velocity -  $x = 6D$ ,  $v$  velocity -  $x = 1.2D$   $u$  velocity,  $x = 1.2D$   $v$  velocity

Figure 4.15 – 2D Inline Cylinder test case - Comparison with literature  
RD schemes and penalization for ice shedding trajectories

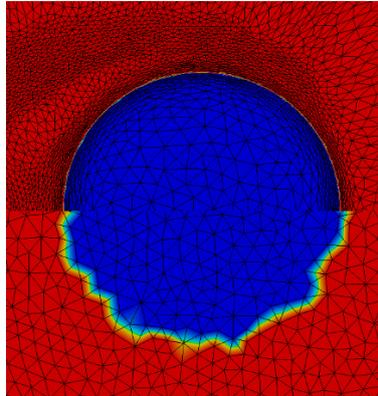


Figure 4.16 – Comparison of solid definition on adapted and initial mesh (same connectivity)

Based on the imposed motion and on the upstream flow conditions, the airfoil experiences an effective angle of attack  $\alpha(t)$  and an effective upstream velocity  $V_{eff}(t)$  defined by

$$\begin{cases} \alpha(t) = \arctan(-V_y(t)/U_\infty) - \theta(t) \\ V_{eff}(t) = \sqrt{(U_\infty^2 + V_y^2(t))}, \end{cases} \quad (4.82)$$

where the freestream velocity far upstream of the oscillating airfoil is  $U_\infty = 68.1$  ( $Ma = 0.2$ ).

A regime corresponding to the parameters  $Re = \frac{U_\infty c}{\nu} = 1100$ ,  $H_0/c = 1$ ,  $f = 0.14$ ,  $x_p/c = 1/3$  and  $\theta_0 = 76.33^\circ$  has been computed. The computational domain is of size  $[-3, 5.5] \times [-6, 6]$  with a finer area of size  $[-1, 4] \times [-2.2, 2.2]$ . It leads to a mesh composed with 30115 vertices and 60186 elements, presented with the sizes for the generation on figure 4.17.

For this test case, elasticity based mesh adaptation is performed, and the monitor function is defined by the following coefficients :  $\alpha_\psi = \alpha_v = 40$ ,  $\alpha_\kappa = 10$ ,  $\beta_\psi = 200$  and  $\beta_v = 2.$ , the physical adaptation being done according to the vorticity. The number of iterations in the Gauss Seidel resolution of the mesh adaptation process is set to 20. Figure 4.19 proposes the vorticity and the corresponding adapted mesh at different times. As for the oscillating cylinder, it is observed that the proposed approach allows to have a refinement close to the solid interface, that imposes accurately the BCs, and that the physics is also precisely resolved in the wanted area thanks to the mesh adaptation.

To validate this test case, aerodynamical coefficients are compared to the forces predictions presented by Kinsey et al. [101], by Campobasso et al. [38] and the ones obtained using the Lescape code. The plots can be found on figure 4.18. It can be seen that the results are in good agreement with the literature.

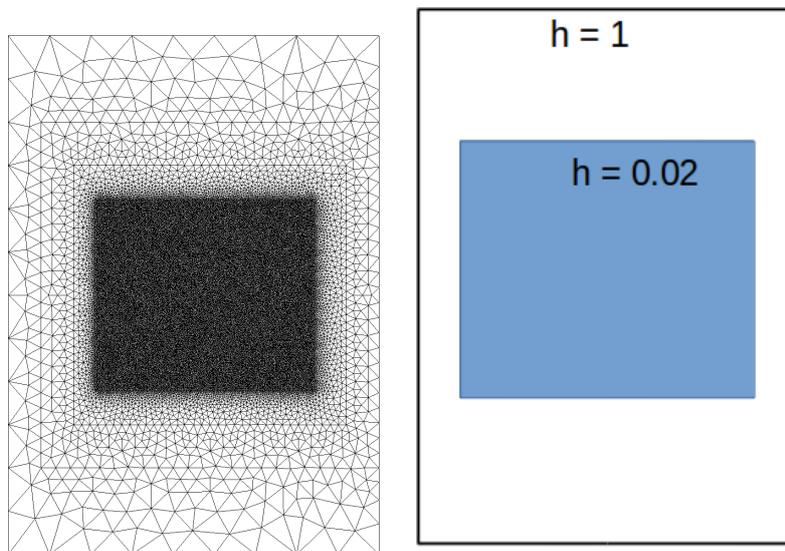


Figure 4.17 – Oscillating Naca test case - Referential mesh

Those results allow to validate, the proposed approach, especially concerning the forces computation. The remaining step is thus to perform simulations where the velocity is this time a resultant of the external forces on the solid to validate the whole process. This is the focus of the next test case.

#### 4.3.2.3 2D Falling Cylinder

This test case simulates the fall of a 2D cylinder (radius  $r = 0.125$ ) on a flat plate in a square cavity of size  $[0, 2] \times [0, 6]$ . The barycentre of the cylinder is initially located in  $(1, 4)$ . The forces influencing its velocity are the aerodynamical forces  $\mathbf{F}$  and the gravity  $\mathbf{G} = (0, -980)$ . The viscosity is set to  $\nu = 0.01$ . The solid and fluid density are respectively  $\rho_s = 1.5$  and  $\rho_f = 1$ . Following [27], equation (4.76) takes the form :

$$\frac{\partial \mathbf{u}_s}{\partial t} = \frac{\mathbf{F}}{m_a} + \frac{\rho_s - \rho_f}{\rho_s} \mathbf{G} \quad (4.83)$$

To impose wall boundary condition on each wall of the cavity, we propose to extend the domain on each side of a size 0.5 and to penalize those added areas (see figure 4.20). The referential mesh is displayed figure 4.20 along with the sizes imposed to generate it. It is composed of 23555 vertices and 46908 elements. The elasticity based adaptation is performed only considering the level set with  $\alpha_\psi = 35$  and  $\beta_\psi = 350$ , a zoom close to the interface on the initial mesh being presented figure 4.21. The ALE CN RD scheme is employed with a CFL equal to 50 and the number of iterations in the Newton Gauss Seidel resolution for mesh adaptation is 15. Figure 4.22 shows the vorticity at time  $t = 0.3125$ .

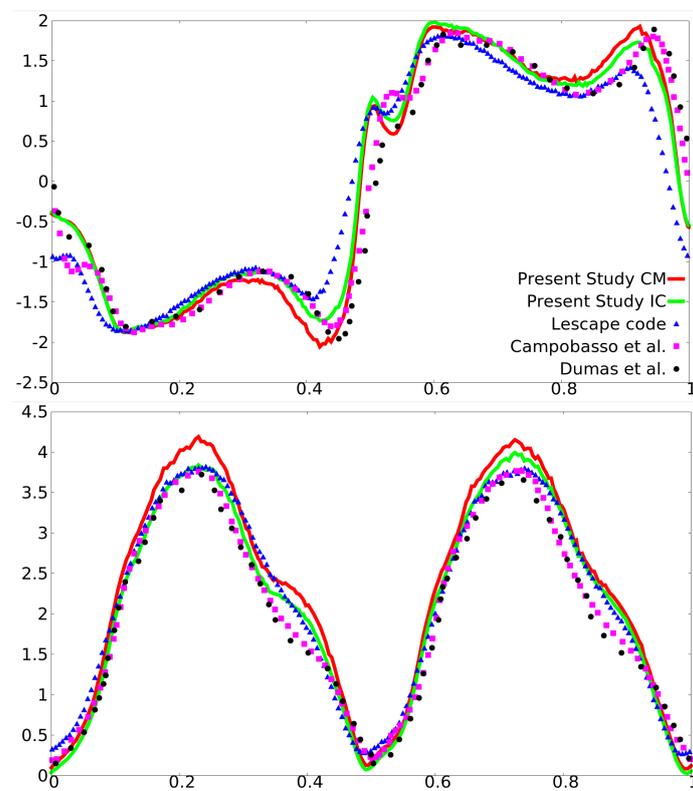


Figure 4.18 – Oscillating Naca0015 airfoil - Aerodynamical coefficients. Top : Lift Coefficient - Bottom : Drag Coefficient

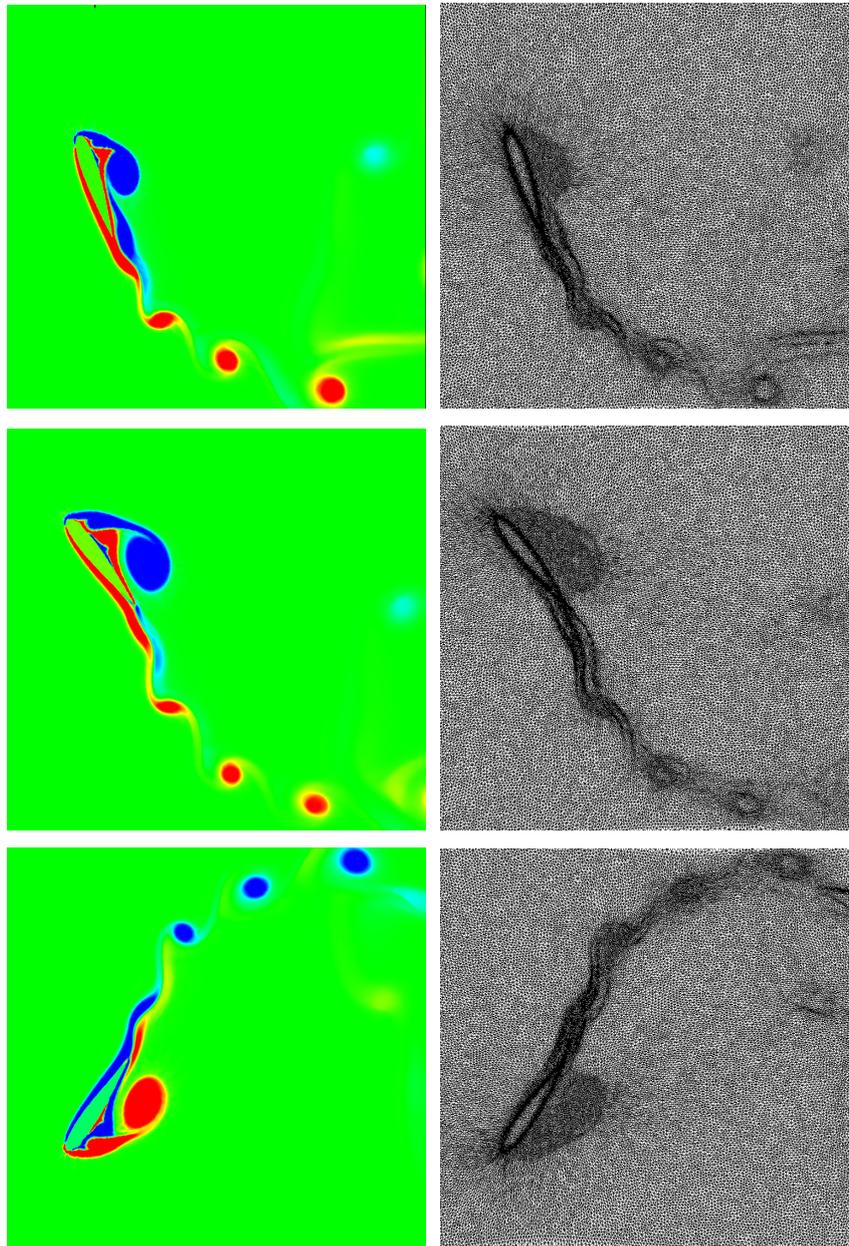


Figure 4.19 – Oscillating naca0015 - Rotational of the velocity and corresponding adapted mesh at different times.

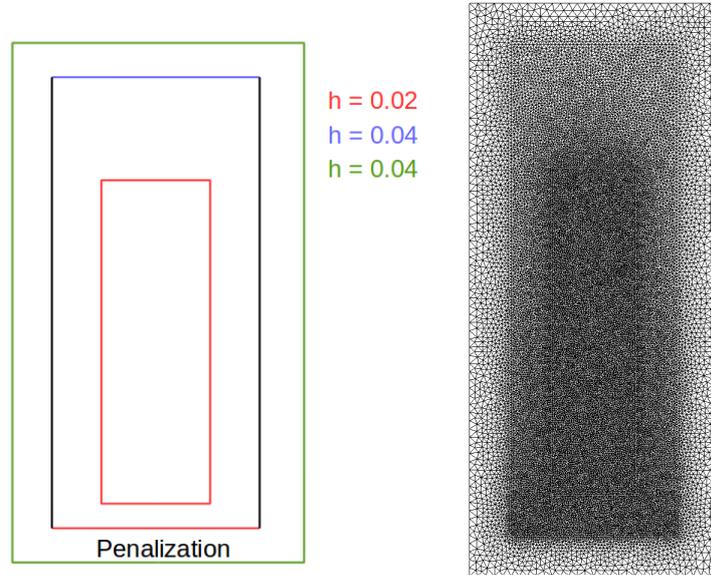


Figure 4.20 – Falling Cylinder - Referential mesh and imposed sizes

This test case has been studied in the literature with penalization on cartesian grids. We compare the evolution of the velocity across time with the results of Glowinski *et. al.* [74], Coquerelle and Cottet [48] and Beaugendre and Morency (Lescape code) [27]. The grids used in the previous references are the same : uniform cartesian grid containing nearly 780000 vertices. For this study, we want to emphasize the accuracy provided by the mesh adaptation. Thus, in addition, the simulation is performed on the referential mesh without performing the adaptation. The curves are plotted figure 4.23. First, we can see that the results provided within this study compares well to the literature and that the approach is consequently validated. Secondly, looking at the steady state of the velocity reached by the cylinder when the frictions compensate the gravity, we can see that for the non adapted simulations, oscillations appear. They are probably due to the lack of definition in the interface, that leads to two main problems : the mass of the solid is not constant along the simulations, and the definition of the interface being not accurate enough, the interpolation procedure from the background mesh to the surfacic mesh is impacted and induces oscillations in the forces computations.

This test case validates the whole FSI process proposed in this work. Now, we are willing to apply it to the context of ice shedding trajectories. Thus, the next and final proposed simulation corresponds with the study of an ice shape put into motion by a flow field.

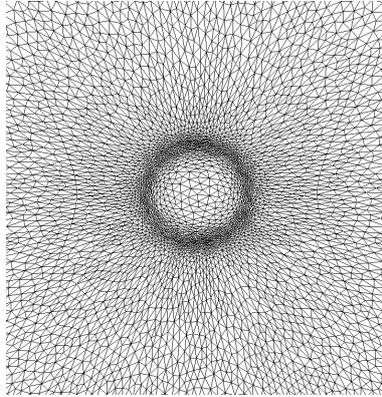


Figure 4.21 – Falling Cylinder - Adapted initial mesh

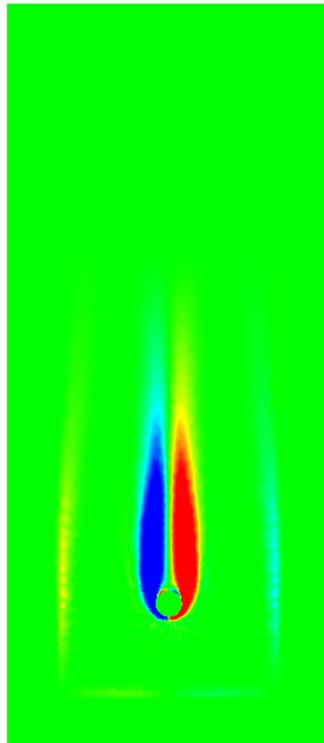


Figure 4.22 – FallingCylinder - Vorticity,  $t = 0.3215$

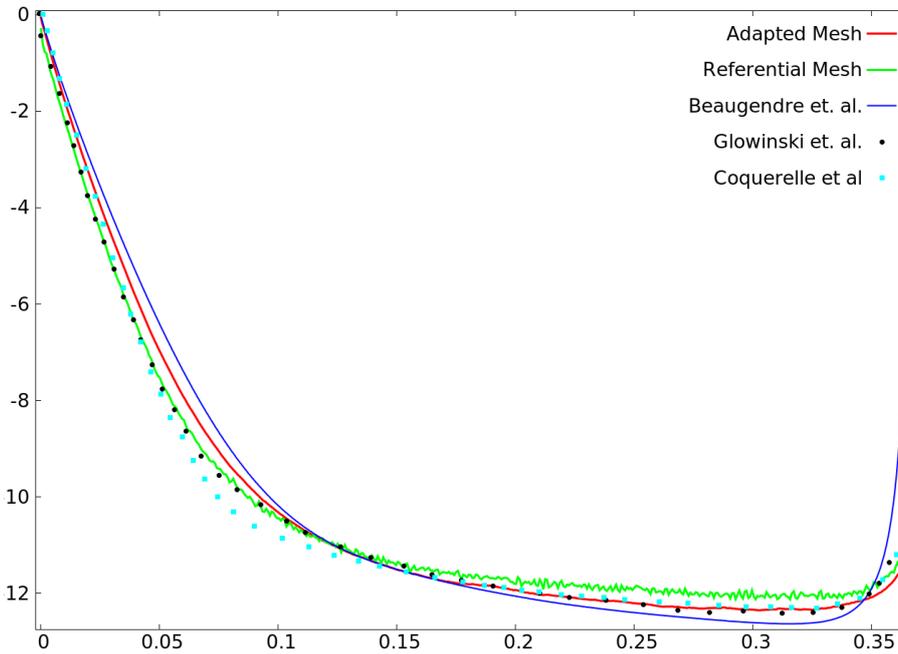


Figure 4.23 – Falling Cylinder - evolution of the solid velocity w.r.t. time

#### 4.3.2.4 GLC305-rime ice shape

In the context of the STORM project, numerical tools should be confronted to experimental data performed at the DLR [100]. The *GLC305-rime* ice shape of chord  $c = 0.1062m$  is employed to perform the study (see figure 4.24 for the 2D cut employed here). Experimentally, the Reynolds number is set to  $Re = 306400$  with an inflow velocity of  $u = 40m/s$ . The ice shape sizing corresponds to real ice thickness accreted on aircraft surface. Experimentally, and in the numerical simulations performed by DLR [100] and ONERA using Chimera method, the ice shape describes a motion characterized by a damped oscillation around a mean pitch angle of  $-45$  deg, and quasi linear trajectory.

However, with such a Reynolds number, a turbulence model is required to perform numerical study. Thus, the settings of the simulation proposed here does not corresponds to the experimental ones as we perform the simulation with a lower Reynolds number :  $Re = 30640$ . In addition, we do not study on the same time interval and computational domain that are smaller for the present simulation. All the other parameters are kept identical to the experimental database. The density of the ice and the flow are respectively  $\rho_s = 920kg/m^3$  and  $\rho_f = 1.2kg/m^3$ , and the shape is submitted to a gravity of  $\mathbf{G} = (0, -9.81)^T m/s^2$ . The inertia momentum is  $M_i = 4,987.10^{-4}kgm^2$ .

This case presents a real challenge for the mesh adaptation process because of the sharp angles and the thin thickness of the ice shape. The initial position of the rotation center is located at  $(0,0)$  in a computational domain of sizes

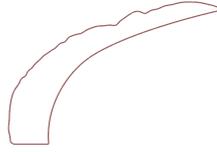


Figure 4.24 – GLC305 rime ice shape

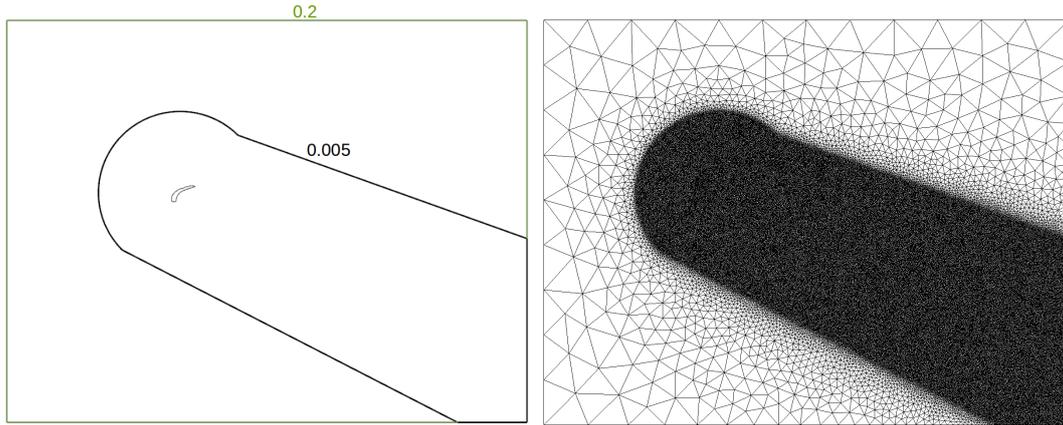


Figure 4.25 – GLC305 simulation - domain and referential mesh

$[-0.75, 1.5] \times [-1, 0.75]$ . Figure 4.25 proposes the domain and initial position of the ice shape, with the refined area in which the solid is expected during the simulation. The imposed sizes and resulting (referential) mesh are also plotted on the same figure.

For this study, the Laplacian based adaptation is employed with the coupling of the SDF and vorticity adaptation. The monitor function associated to the 0 level set is defined constant by part (4.73) with the following parameters :  $w_1 = 0.003$ ,  $w_2 = 0.06$ ,  $w_3 = 0.15$  and  $C_1 = 1120$ ,  $C_2 = 275$ ,  $C_3 = 160$ ,  $C_4 = 100$  and the adaptation to the vorticity is performed using  $\beta_v = 3.5$ . For the definition of the global monitor function (4.60), we set  $\epsilon = 0.005$ . The initial mesh (only adapted to the level set) is displayed figure 4.26 with a comparison against the referential mesh and 0 isovalue of the level set. Looking at this figure 4.26, the ability of the mesh adaptation to lighten the number of nodes is emphasized. On the referential mesh, in addition of a poor description of the 0 level set, only a few nodes define the inside of the solid. On the contrary, the adapted mesh proposes an accurate description of the solid and enough points inside the solid to impose accurately the penalization.

The first step is to solve on this initial adapted mesh the equations till the establishment of a flow before allowing the motion. Figure 4.27 corresponds to this state, that is the initial condition for the fall of the ice shape. Once the ice is free, the aerodynamical forces and the gravity rule the movement with equation (4.83). The ALE CN RD scheme is employed with a CFL of 50 and

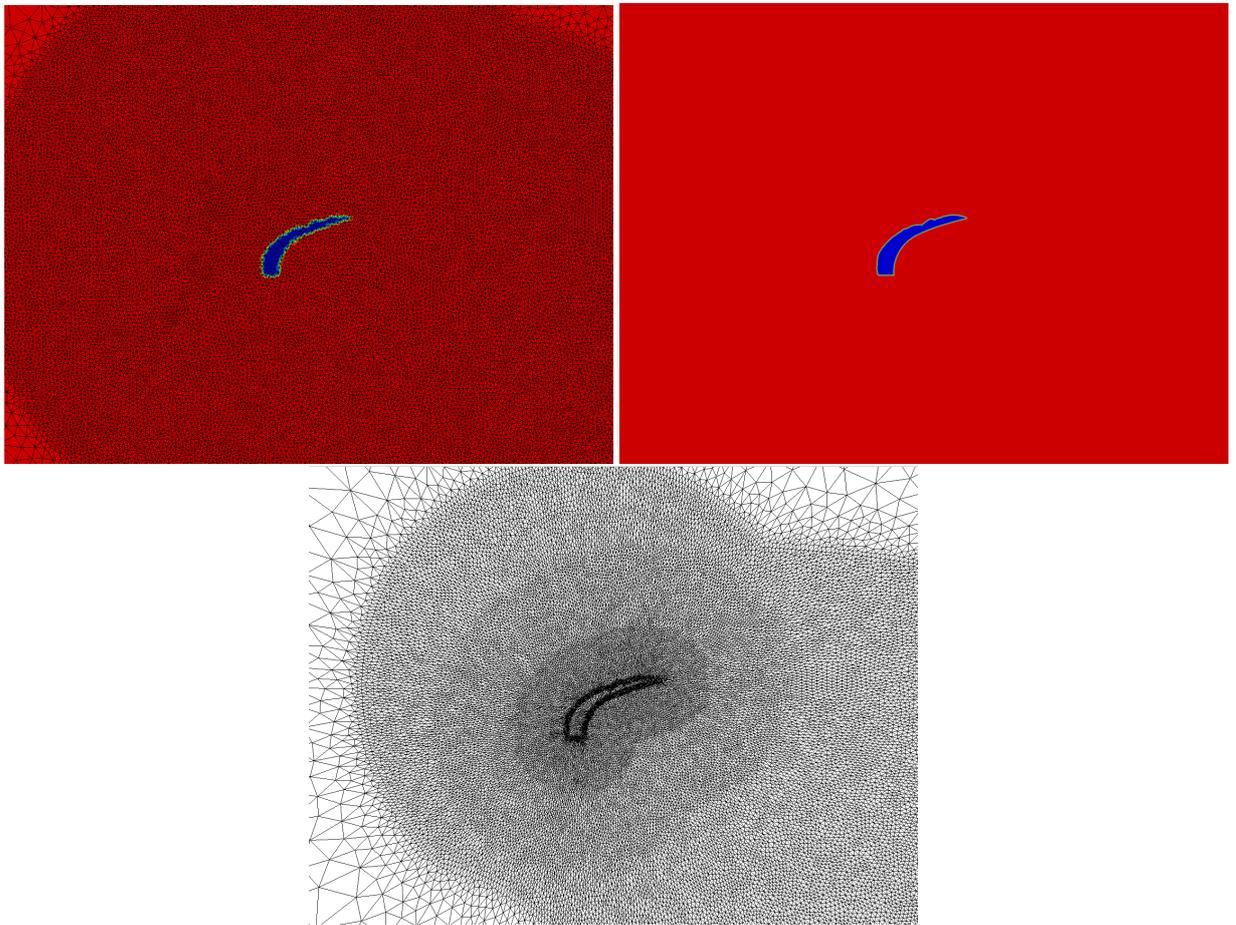


Figure 4.26 – GLC305 Adaptation. Top Left : 0 level set on referential mesh  
 - Top Right : 0 level set on adapted mesh - Bottom : Adapted mesh

40 Newton Jacobi iterations are performed for the mesh adaptation.

Figure 4.28 displays the position of the ice shape according to the time. Even if the settings differ from the experiment, we also recover an oscillating behaviour of the ice, along with a nearly linear trajectory. Figures 4.29 propose the vorticity at different time steps and a zoom close to the interface to prove the ability of the proposed mesh adaptation technique to keep the same accuracy all along the simulation.

Even if this simulation cannot be validated with comparison to other computational codes or experimental data, it shows the ability provided by the proposed combination IB/mesh adaptation to handle large displacement and can be applied to ice shedding simulations. We obtain an expected behaviour from the ice shape concerning the linear trajectory and oscillations around a mean pitch angle that have been observed experimentally and that was sought, even for lower Reynolds.

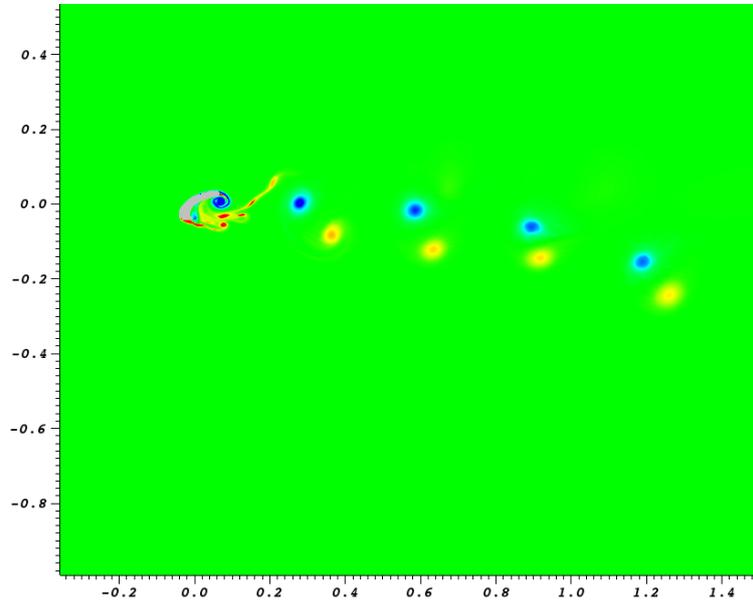


Figure 4.27 – Vorticity at the beginning of the GLC305 rime ice motion

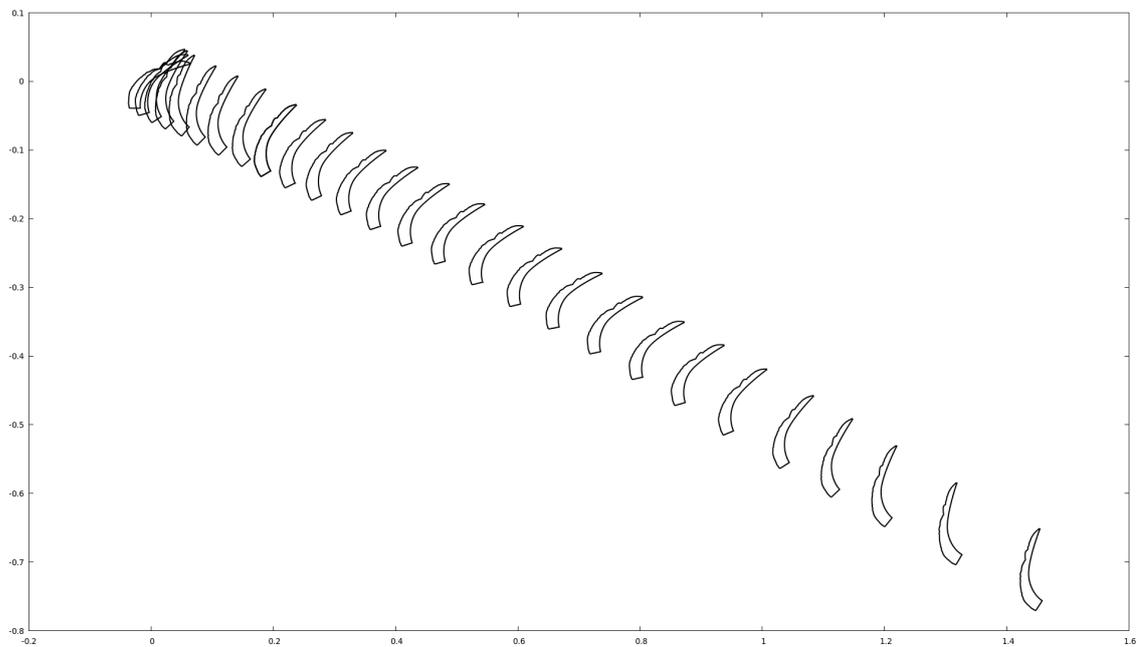


Figure 4.28 – GLC305 surface isoline at different times

## 4.4 Summary

This final chapter demonstrated the relative ease provided by the penalization to perform the simulations of moving solids into a fluid. The unsteady schemes proposed chapter 3 have been extended to the ALE framework and coupled to an r-adaptivity mesh adaptation. Such a procedure involves the resolution of an elasticity problem at each time step, that is solved using a limited number of Newton iterations. It allows to provide an adaptation to the solid boundary all along the simulation, so as to avoid loss of accuracy in the imposition of the BC *via* the penalization, and to perform sufficiently accurate evaluations of the aerodynamical forces. Test cases where the solid motion is ruled by an analytical function have been used to validate the process and the forces computations. A last validation test case involving a solid motion ruled by external forces has validated the full FSI process proposed, that is a combination of the different tools developed during this work. Finally, this technique has been applied to a GLC305 rime ice shape, one of the study of the STROM project. Even if this is only a preliminary results, it comforts the idea of using IBM/mesh adaptation for such problems as the behaviour is the one expected.

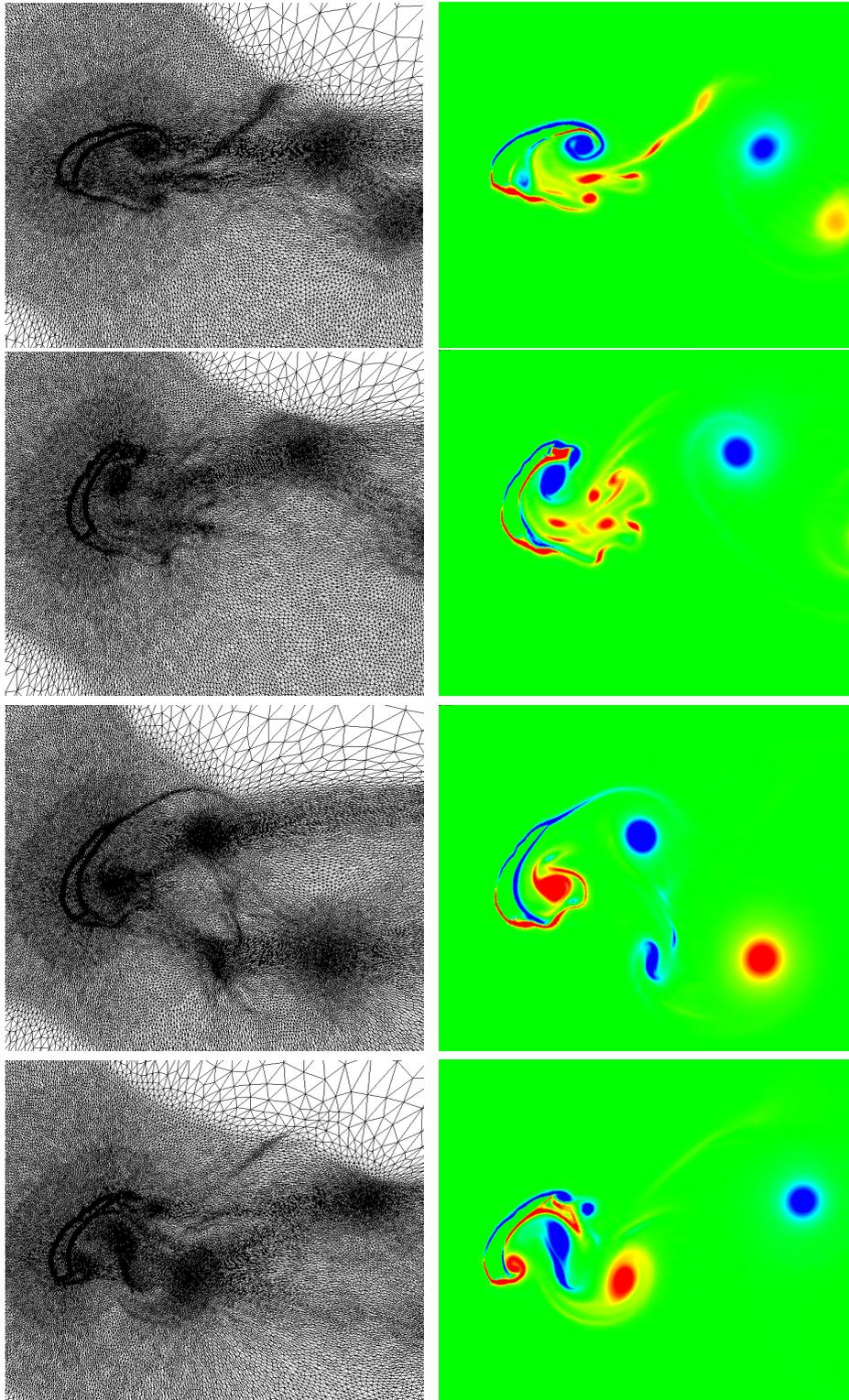


Figure 4.29 – Zoom on the mesh at times  $t = 0$ ,  $t = 0.027$ ,  $t = 0.055$  and  $t = 0.083$



# Chapter 5

## Conclusion

---

## Conclusion

In this work, the simulations using an immersed boundary method called penalization on adapted unstructured grids with a residual distribution framework have been addressed. First, to demonstrate the benefit of the combination IB/unstructured grids, steady simulations on adapted mesh increasing the definition of the solid and the physical solution have been proposed. The discretization of the equations has been done using the residual distribution framework. Thus, the considered schemes have been extended to the penalized Navier Stokes equations by a simple Galerkin discretization of the penalty term. Different simulations have been proposed to prove that such schemes are well suited for those simulations and that the proposed methodology presents a competitive alternative to classical body fitted simulations [123, 3].

The study has then been extended to unsteady simulations, still considering fixed grids, for motionless solids. To perform such simulations, the schemes have been extended to unsteady advection diffusion problems and the penalization has been handled by the use of a splitting technique [124]. This approach provides some freedom in the choice of the time discretization scheme. Two time discretizations have been developed to solve the unsteady Navier Stokes equations : a second order explicit Runge Kutta scheme and an implicit Crank Nicolson scheme. To avoid a loss of accuracy with the splitting approach, the Strang splitting has been used, leading to simple ordinary differential equations for the treatment of the penalty part, and thus a nodal imposition of the boundary conditions.

As the aim is to handle objects in motion, the necessity of an unsteady adaptation technique is required. To avoid remeshing/interpolation procedures, an r-adaptivity mesh strategy combined to an ALE resolution of the equations has been proposed. Based on a monitor function defining areas of refinement according to the position of the solid, and if wanted a chosen physical variable at each time step, the mesh is adapted without changing its connectivity. The preliminary results obtained prove the competitive aspect of such an approach for considering moving objects [125]. This approach provides a good implicit definition of the IB, even if the background mesh is not very fine, as the adaptation allows to refine close to the geometry all along the simulation. In addition, there is no need to remesh the computational domain and to use interpolation procedures. Once the approach has been validated on several test cases, it has been applied to the simulation of an ice shape trajectory. Even if no proper validation can be provided for this last test case, the results are the ones expected, further study being required for a complete validation.

This work has provided an innovative and promising approach for the simulation of fluid structure interaction problems, that should lead to an efficient high fidelity model for ice shedding trajectories prediction regarding the pre-

liminary results obtained in this context. The focus needs to be put on some specific aspects (listed in the next section) to perform complete ice shedding studies.

## Perspective and On going work

- **3D Motion simulation / r-adaptation parallelization :** The first aspect that should be mentioned is the necessity of the parallelization of the r-adaptation process to be able to perform 3D simulations in the context of moving bodies. Indeed, even if the adopted strategy allows to perform mesh adaptation with a limited computational time, the procedure in 3 dimensions still requires a lot of time. By lack of time, the current implementation is sequential, which remains suitable for 2D simulations (the fluid resolution being parallel), but is a real inconvenient in 3D. The parallelization of the proposed approach is not a difficult task due to the constant node connectivity which provides a constant load balancing.
- **Turbulence Model :** Concerning the ice shedding problems, a missing tool that is unavoidable because of the high Reynolds number involved is a turbulence modelling. From a theoretical point of view, a model such as the Spalart-Allmaras one should be well adapted to the penalization problem because BC are easier for this turbulence model. Nevertheless, some points will need a specific focus, especially the generation of a mesh sufficiently adapted to the problem (some wall law being a possibility to lighten a little such constraint).
- **Forces computations and FSI simulations :** The forces computation employed during this work has provided good results and allowed to performed full FSI simulations. However, to get better results, especially concerning turbulent simulations, improvement should be investigated. Some ideas could be to perform better gradient reconstruction or improve the interpolation procedure from the background mesh to the surfacic one. In addition, it could be interesting to look at different resolution of the equations ruling the motion, as only Euler discretization are employed to evaluate the solid velocity from the forces.
- **r-adaptation improvement :** As demonstrated in this manuscript, the r-adaptation strategy allows to improve the definition of the IB on the background mesh. However, as mentioned in the previous point, a turbulence model being required, a task will be to provide meshes adapted for those simulations. Investigation could be done concerning the definition of the monitor function, or an improvement of the problem

---

to solve. Combination between Laplacian and elasticity based approach could be a path to investigate.

In addition, such an adaptation cannot be as accurate as h-adaptation techniques that provide almost optimal meshes for the considered problem. However, it is not obvious that such an accuracy is mandatory for the involved application, as many other factors intervene, that could spoil this accuracy. It would be interesting to compare the two approaches, especially to compare accuracy w.r.t. computational time.

The last point to mention about r-adaptation concerns the number of nodes necessary for the simulations. Indeed, if the point emphasized is the non necessity to perform remeshing and interpolations, to perform the r-adaptation w.r.t. the solid position, the original background mesh must provide a finer area according to the displacement of the object. It can be easily imagined that in the context of ice shedding trajectories, owing to the random character of the movement and the very high sensibility to initial conditions, all the possible trajectories embrace a significant domain. The refined area should be in coherence with this domain and might lead to an initial quite heavy mesh. An investigation should be done to couple r-adaptation and h-adaptation to provide the refined area only where necessary, as performed for the moving mesh techniques with explicit discretization of the solid presented in the introduction of this manuscript [1.2.2.1](#). However, in comparison with those techniques, the number of remeshing steps should be a lot reduced, as the mesh would be suited for larger time step.

# Appendix A

## Advection and Diffusion Tensors for Navier-Stokes Equations

### A.1 Euler Jacobian

First let recall the expression of the Euler flux  $\mathbf{F}^{Eul}$  for a 3 dimensional problem :

$$\mathbf{F}^{Eul} = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} - p \mathbb{I} \\ (\rho e + p) \mathbf{v} \end{pmatrix} \quad (\text{A.1})$$

The three components are :

$$\mathbf{F}_x^{Eul} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ \rho v_x v_z \\ (\rho e + p) v_x \end{pmatrix}, \mathbf{F}_y^{Eul} = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + p \\ \rho v_y v_z \\ (\rho e + p) v_y \end{pmatrix}, \mathbf{F}_z^{Eul} = \begin{pmatrix} \rho v_z \\ \rho v_x v_z \\ \rho v_y v_z \\ \rho v_z^2 + p \\ (\rho e + p) v_z \end{pmatrix} \quad (\text{A.2})$$

The computation of the  $\mathbb{A}_x = \partial \mathbf{F}_x^{Eul} / \partial \mathbf{u}$  is presented, the obtaining of the two other ones being completely analogue.

First, let rewrite the vector as dependant of the conservative variable  $\rho$ ,  $m_x = \rho v_x$ ,  $m_y = \rho v_y$ ,  $m_z = \rho v_z$ ,  $m_e = \rho e$ . Using the relations  $p = \rho(\gamma - 1)\epsilon$  coming from the PG law and  $e = \epsilon + \frac{\|\mathbf{u}\|^2}{2}$ , it can be obtained :

$$\mathbf{F}_x^{Eul} = \begin{pmatrix} \frac{m_x^2}{\rho} - \frac{\gamma-1}{2} \frac{m_x^2 + m_y^2 + m_z^2}{\rho} + (\gamma - 1) m_e \\ \frac{m_x m_y}{m_x m_z} \\ \frac{\rho}{m_x m_z} \\ \gamma \frac{m_e m_x}{\rho} - \frac{\gamma-1}{2} \frac{m_x}{\rho^2} (m_x^2 + m_y^2 + m_z^2) \end{pmatrix} \quad (\text{A.3})$$

After some calculation, it is obtained :

$$\mathbb{A}_x = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - v_x^2 & (3 - \gamma)v_x & -(\gamma - 1)v_y & -(\gamma - 1)v_z & \gamma - 1 \\ -v_x v_y & v_y & v_x & 0 & 0 \\ -v_x v_z & v_z & 0 & v_x & 0 \\ v_x \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - \alpha \right] & -(\gamma - 1)v_x^2 + \alpha & -(\gamma - 1)v_x v_y & -(\gamma - 1)v_x v_z & \gamma v_x \end{pmatrix}$$

where  $\alpha = \frac{c^2}{\gamma - 1} + \frac{\|\mathbf{v}\|^2}{2}$ . Some details are given for the derivation of the energy component of  $\mathbf{F}_x^{Eul}$  with respect to  $\rho$ :

$$\begin{aligned} \frac{\partial F_{x,5}^{Eul}}{\partial \rho} &= -\gamma v_x e + (\gamma - 1)v_x \|\mathbf{v}\|^2 \\ &= v_x \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} + (\gamma - 1) \left( \frac{\|\mathbf{v}\|^2}{2} - e \right) - e \right] \\ &= v_x \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - \left( \frac{p}{\rho} + \epsilon + \frac{\|\mathbf{v}\|^2}{2} \right) \right] \\ &= v_x \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - \alpha \right] \quad \text{using PG law : } c = \sqrt{\frac{\gamma p}{\rho}} \text{ and } \epsilon = \frac{c^2}{\gamma(\gamma - 1)} \end{aligned}$$

Identically, it is get :

$$\mathbb{A}_y = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ -v_x v_y & v_y & v_x & 0 & 0 \\ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - v_y^2 & -(\gamma - 1)v_x & (3 - \gamma)v_y & -(\gamma - 1)v_z & \gamma - 1 \\ -v_y v_z & 0 & v_z & v_y & 0 \\ v_y \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - \alpha \right] & -(\gamma - 1)v_x v_y & -(\gamma - 1)v_y^2 + \alpha & -(\gamma - 1)v_y v_z & \gamma v_y \end{pmatrix}$$

$$\mathbb{A}_z = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ -v_x v_z & v_z & 0 & v_x & 0 \\ -v_y v_z & 0 & v_z & v_y & 0 \\ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - v_z^2 & -(\gamma - 1)v_x & -(\gamma - 1)v_y & (3 - \gamma)v_z & \gamma - 1 \\ v_z \left[ (\gamma - 1)\frac{\|\mathbf{v}\|^2}{2} - \alpha \right] & -(\gamma - 1)v_x v_z & -(\gamma - 1)v_y v_z & -(\gamma - 1)v_z^2 + \alpha & \gamma v_z \end{pmatrix}$$

## A.2 Navier Stokes Tensor

Let interest on the  $x$  component of the flux function  $T_x^{NS}$  expressed as dependant of the conservative variables  $\rho, m_x, m_y, m_e$  and its associated component of  $\mathbb{K}$ .

$$F_x^{NS} = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \frac{m_x}{\rho} \sigma_{xx} + \frac{m_y}{\rho} \sigma_{xy} + \frac{m_z}{\rho} \sigma_{xz} - q_x \end{pmatrix}$$

## A. Advection and Diffusion Tensors for Navier-Stokes Equations

---

Using PG law, it is set  $T = \frac{\gamma-1}{\mathcal{R}} \left( e - \frac{\|\mathbf{v}\|^2}{2} \right)$  which allows to rewrite  $q_x = -\kappa \frac{\partial T}{\partial x}$  as :

$$q_x = -\kappa \frac{\gamma-1}{\mathcal{R}} \frac{1}{\rho} \left[ \left( \frac{\partial m_e}{\partial x} - \frac{m_e}{\rho} \frac{\partial \rho}{\partial x} \right) - \left( \frac{m_x}{\rho} \frac{\partial m_x}{\partial x} - \frac{m_x^2}{\rho^2} \frac{\partial \rho}{\partial x} + \frac{m_y}{\rho} \frac{\partial m_y}{\partial x} - \frac{m_y^2}{\rho^2} \frac{\partial \rho}{\partial x} + \frac{m_z}{\rho} \frac{\partial m_z}{\partial x} - \frac{m_z^2}{\rho^2} \frac{\partial \rho}{\partial x} \right) \right]$$

Now, denoting  $x_1 = x$ ,  $x_2 = y$ ,  $x_3 = z$ ,  $v_1 = v_x$ ,  $v_2 = v_y$ ,  $v_3 = v_z$ , the components of the stress tensor can be written :

$$\begin{aligned} \sigma_{ij} &= \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \delta_{ij} \frac{2}{3} \sum_{k=1}^3 \frac{\partial v_k}{\partial x_k} \right) \\ &= \frac{\mu}{\rho} \left[ \frac{\partial m_i}{\partial x_j} - \frac{m_i}{\rho} \frac{\partial \rho}{\partial x_j} + \frac{\partial m_j}{\partial x_i} - \frac{m_j}{\rho} \frac{\partial \rho}{\partial x_i} - \delta_{ij} \frac{2}{3} \sum_{k=1}^3 \left( \frac{\partial m_k}{\partial x_k} - \frac{m_k}{\rho} \frac{\partial \rho}{\partial x_k} \right) \right] \end{aligned}$$

with  $\delta_{ij}$  the Kronecker delta.

It leads to :

$$\begin{aligned} F_{x1}^{NS} &= 0 \\ F_{x2}^{NS} &= \frac{\mu}{\rho} \left[ \frac{4}{3} \left( \frac{\partial m_x}{\partial x} - \frac{m_x}{\rho} \frac{\partial \rho}{\partial x} \right) - \frac{2}{3} \left( \frac{\partial m_y}{\partial y} - \frac{m_y}{\rho} \frac{\partial \rho}{\partial y} \right) - \frac{2}{3} \left( \frac{\partial m_z}{\partial z} - \frac{m_z}{\rho} \frac{\partial \rho}{\partial z} \right) \right] \\ F_{x3}^{NS} &= \frac{\mu}{\rho} \left( \frac{\partial m_x}{\partial y} - \frac{m_x}{\rho} \frac{\partial \rho}{\partial y} + \frac{\partial m_y}{\partial x} - \frac{m_y}{\rho} \frac{\partial \rho}{\partial x} \right) \\ F_{x4}^{NS} &= \frac{\mu}{\rho} \left( \frac{\partial m_x}{\partial z} - \frac{m_x}{\rho} \frac{\partial \rho}{\partial z} + \frac{\partial m_z}{\partial x} - \frac{m_z}{\rho} \frac{\partial \rho}{\partial x} \right) \\ F_{x5}^{NS} &= \frac{m_x}{\rho} F_{x2}^{NS} + \frac{m_y}{\rho} F_{x3}^{NS} + \frac{m_z}{\rho} F_{x4}^{NS} + \\ &\quad \kappa \frac{\gamma-1}{\mathcal{R}} \frac{1}{\rho} \left[ \left( \frac{\partial m_e}{\partial x} - \frac{m_e}{\rho} \frac{\partial \rho}{\partial x} \right) - \left( \frac{m_x}{\rho} \frac{\partial m_x}{\partial x} - \frac{m_x^2}{\rho^2} \frac{\partial \rho}{\partial x} + \frac{m_y}{\rho} \frac{\partial m_y}{\partial x} - \frac{m_y^2}{\rho^2} \frac{\partial \rho}{\partial x} + \frac{m_z}{\rho} \frac{\partial m_z}{\partial x} - \frac{m_z^2}{\rho^2} \frac{\partial \rho}{\partial x} \right) \right] \end{aligned}$$

From those expression, the matrices  $\underline{K}_{11}$ ,  $\underline{K}_{12}$  and  $\underline{K}_{13}$  can be identified :

$$\underline{K}_{11} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{4}{3}v_x & \frac{4}{3} & 0 & 0 & 0 & 0 \\ -v_y & 0 & 1 & 0 & 0 & 0 \\ -v_z & 0 & 0 & 1 & 0 & 0 \\ -\frac{v_x^3}{3} - \|\mathbf{v}\|^2 - \alpha(e - \|\mathbf{v}\|^2) & (\frac{4}{3} - \alpha)v_x & (1 - \alpha)v_y & (1 - \alpha)v_z & \alpha & \alpha \end{pmatrix}$$

with  $\alpha = \frac{\kappa(\gamma-1)}{\mu\mathcal{R}}$ .

$$\underline{K}_{13} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\frac{2}{3}v_y & 0 & -\frac{2}{3} & 0 & 0 \\ -\frac{2}{3}v_x & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{3}v_x v_y & v_y & -\frac{2}{3}v_x & 0 & 0 \end{pmatrix}, \underline{K}_{12} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}v_z & 0 & 0 & -\frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -v_x & 1 & 0 & 0 & 0 \\ -\frac{1}{3}v_x v_z & v_z & 0 & -\frac{2}{3}v_x & 0 \end{pmatrix}$$

Performing the same analysis with  $F_y^{NS}$  and  $F_z^{NS}$  lead to the other matrix components of  $\mathbb{K}$  :

$$\underline{K}_{22} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -v_x & 1 & 0 & 0 & 0 \\ -\frac{4}{3}v_y & 0 & \frac{4}{3} & 0 & 0 \\ -v_z & 0 & 0 & 1 & 0 \\ -\frac{v_y^2}{3} - \|\mathbf{v}\|^2 - \alpha(e - \|\mathbf{v}\|^2) & (1-\alpha)v_x & (\frac{4}{3}-\alpha)v_y & (1-\alpha)v_z & \alpha \end{pmatrix}$$

$$\underline{K}_{21} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -v_y & 0 & 1 & 0 & 0 \\ \frac{2}{3}v_x & -\frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{3}v_xv_y & -\frac{2}{3}v_y & v_x & 0 & 0 \end{pmatrix}, \underline{K}_{23} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}v_z & 0 & 0 & -\frac{2}{3} & 0 \\ -v_y & 0 & 1 & 0 & 0 \\ -\frac{1}{3}v_yv_z & 0 & v_z & -\frac{2}{3}v_y & 0 \end{pmatrix}$$

$$\underline{K}_{33} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -v_x & 1 & 0 & 0 & 0 \\ -v_y & 0 & 1 & 0 & 0 \\ -\frac{4}{3}v_z & 0 & 0 & \frac{4}{3} & 0 \\ -\frac{v_z^2}{3} - \|\mathbf{v}\|^2 - \alpha(e - \|\mathbf{v}\|^2) & (1-\alpha)v_x & (1-\alpha)v_y & (\frac{4}{3}-\alpha)v_z & \alpha \end{pmatrix}$$

$$\underline{K}_{31} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -v_z & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{3}v_x & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{1}{3}v_xv_z & -\frac{2}{3}v_z & 0 & v_x & 0 \end{pmatrix}, \underline{K}_{33} = \frac{\mu}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -v_z & 0 & 0 & 1 & 0 \\ -\frac{2}{3}v_y & 0 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3}v_yv_z & 0 & -\frac{2}{3}v_z & v_y & 0 \end{pmatrix}$$

# Bibliography

- [1] R. Abgrall. Toward the ultimate conservative scheme: Following the quest. *J Comput Phys*, 167(2):277 – 315, 2001.
- [2] R. Abgrall. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. *J Comput Phys*, 214(2):773 – 808, 2006.
- [3] R. Abgrall, H. Alcin, H. Beaugendre, C. Dobrzynski, and L. Nouveau. Residual schemes applied to an embedded method expressed on unstructured adapted grids. In *Conference on Computational Fluid Dynamics (ICCFD8)*, number ICCFD8-2014-0166, Chengdu, China, July 2014.
- [4] R. Abgrall, H. Beaugendre, and C. Dobrzynski. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *J Comput Phys*, 257:83–101, 2014.
- [5] R. Abgrall, P.M. Congedo, D. De Santis, and N. Razaaly. A non-linear residual distribution scheme for real-gas computations. *Comput Fluids*, 102:148 – 169, 2014.
- [6] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *J Comput Phys*, 230(11):4103 – 4136, 2011.
- [7] R. Abgrall and K. Mer. Un théorème de type lax-wendroff pour les schémas distributifs. Technical Report 98010, INRIA, Le Chesnay, FRANCE, 1998.
- [8] R. Abgrall and M. Ricchiuto. *High order methods for CFD*. John Wiley & Sons, to appear.
- [9] R. Abgrall and P.L. Roe. High order fluctuation schemes on triangular meshes. *J Sci Comput*, 19:0885–7474, 2003.
- [10] R. Abgrall, D. De Santis, and M. Ricchiuto. High order preserving residual distribution schemes for advection-diffusion problems on arbitrary grids. *SIAM J Sci Comput*, 36:A955–A983, 2014.

- [11] R. Abgrall, D. De Santis, and M. Ricchiuto. High-order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. *SIAM J Sci Comput*, 36(3):A955–A983, 2014.
- [12] Rémi Abgrall and Timothy Barth. Residual distribution schemes for conservation laws via adaptive quadrature. *SIAM J Sci Comput*, 24(3):732–769, 2002.
- [13] Rémi Abgrall and Mohamed Mezine. Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. *J Comput Phys*, 188(1):16 – 55, 2003.
- [14] F. Alauzet. Size gradation control of anisotropic meshes. *Finite Elements in Analysis and Design*, 46(1–2):181 – 202, 2010. Mesh Generation - Applications and Adaptation.
- [15] Frédéric Alauzet. A changing-topology moving mesh technique for large displacements. *Eng Comput*, 30(2):175–200, 2014.
- [16] Frédéric Alauzet and David Marcum. A closed advancing-layer method with connectivity optimization-based mesh movement for viscous mesh generation. *Eng Comput*, 31(3):545–560, 2015.
- [17] Frédéric Alauzet and Adrien Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Comput Aided Design*, 72:13 – 39, 2016. 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.
- [18] P. Angot, C-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer Math*, 81:497–520, 1999.
- [19] L. Arpaia and M. Ricchiuto. r-adaptation for shallow water flows: conservation, well balancedness, efficiency. In *VII European Congress on Computational Methods in Applied Sciences & Engineering*, Crete, 2016. Inria RR8956, submitted to *Comput Fluids*.
- [20] L. Arpaia, M. Ricchiuto, and R. Abgrall. An ale formulation for explicit runge-kutta residual distribution. *J Sci Comput*, 63(2):502–547, 2015.
- [21] E. Arquis and J.P. Caltagirone. Sur les conditions hydrodynamiques au voisinage d’une interface milieu fluide - milieu poreux: application a la convection naturelle. *R Acad Sci Paris 299(1), Serie II*, 1-4, 1984.
- [22] C.M.J. Baker, A.G. Buchan, C.C. Pain, P.E. Farrell, M.D. Eaton, and P. Warner. Multimesh anisotropic adaptivity for the boltzmann transport equation. *Annals of Nuclear Energy*, 53:411 – 426, 2013.

## BIBLIOGRAPHY

---

- [23] T. Barth and P. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. In *28th AIAA Aerospace Sciences Meetings*, Reno, Nevada, January 1990.
- [24] Timothy J. Barth. *Numerical Methods for Gasdynamic Systems on Unstructured Meshes*, pages 195–285. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
- [25] Guido S. Baruzzi, Patrick Lagacé, Martin S. Aubé, and Wagdi G. Habashi. Development of a shed-ice trajectory simulation in fensap-ice. In *SAE Technical Paper*. SAE International, 09 2007.
- [26] Héloïse Beaugendre, François Morency, Federico Gallizio, and Sophie Laurens. Computation of ice shedding trajectories using cartesian grids, penalization, and level sets. *Model Simul Eng*, 2011:3:1–3:15, January 2011.
- [27] Héloïse Beaugendre and François Morency. Innovative Model for Flow Governed Solid Motion based on Penalization and Aerodynamic Forces and Moments. Research Report RR-8718, INRIA Bordeaux, équipe CARDAMOM ; IMB ; INRIA, April 2015.
- [28] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *J Comput Phys*, 231(19):6323 – 6348, 2012.
- [29] Michel Bergmann, Laurent Cordier, and Jean-Pierre Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Phys Fluids*, 17(9), 2005.
- [30] R. P. Beyer and R. J. LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J Numer Anal*, 29(2):332–364, 1992.
- [31] Laure Billon, , Youssef Mesri, , and Elie Hachem. Anisotropic boundary layer mesh generation for immersed complex geometries. *Eng Comput*, pages 1–12, 2016.
- [32] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large mach number flows in the presence of obstacles. *Comput Fluids*, 38(3):703 – 714, 2009.
- [33] M. Braza, P. Chassaing, and H. Ha Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *J Fluid Mech*, 165:79–130, 1986.

- [34] H.C. Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Appl Sci Res*, 1(1):27–34, 1949.
- [35] A. N. Brooks and T. J.R. Hughes. Streamline upwind/ Petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp Meth Appl Mech Engrg*, 32(1):199 – 259, 1982.
- [36] C J. Budd, W Huang, and R D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 5 2009.
- [37] Thi Thu Cuc Bui, P. Frey, and B. Maury. A coupling strategy based on anisotropic mesh adaptation for solving two-fluid flows. *Int J Numer Meth Fl*, 66(10):1226–1247, 2011.
- [38] M. S. Campobasso and J. Drofelnik. Compressible Navier-Stokes analysis of an oscillating wing in a power-extraction regime using efficient low-speed preconditioning. *Comput Fluids*, 67:26–40, 2012.
- [39] D. Caraeni and L. Fuchs. *LES using a Parallel Multidimensional Upwind Solver*, pages 461–466. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [40] D. Caraeni and L. Fuchs. Compact third-order multidimensional upwind scheme for Navier-Stokes simulations. *Theor Comp Fluid Dyn*, 15(6):373–401, 2002.
- [41] D.A. Caraeni. *Development of a Multidimensional Upwind Residual Distribution Solver for Large Eddy Simulation of Industrial Turbulent Flows*. PhD thesis, Lund Institute of Technology, 2000.
- [42] Reuben Chandrasekharan and Michael Hinson. Trajectory simulation of ice shed from a business jet. In *SAE Technical Paper*. SAE International, 09 2003.
- [43] G. Chen, H. Tang, and P. Zhang. Second-order accurate Godunov scheme for multicomponent flows on moving triangular meshes. *J Sci Comput*, 34(1):64–86, 2008.
- [44] D. K. Clarke, H. A. Hassan, and M. D. Salas. *Euler calculations for multielement airfoils using Cartesian grids*, volume 24, pages 353–358. AIAA Journal, 1986.
- [45] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J Comput Phys*, 84(1):90 – 113, 1989.

- [46] Bernardo Cockburn and Chi-Wang Shu. The runge-kutta local projection  $p^1$ -discontinuous-galerkin finite element method for scalar conservation laws. *IMA Preprint Series*, (388), 1988.
- [47] Bernardo Cockburn and Chi-Wang Shu. The runge–kutta discontinuous galerkin method for conservation laws v. *J Comput Phys*, 141(2):199 – 224, 1998.
- [48] Mathieu Coquerelle and Georges-Henri Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J Comput Phys*, 227(21):9121–9137, November 2008.
- [49] T. Coupez. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *J Comput Phys*, 230(7):2391 – 2405, 2011.
- [50] T. Coupez and E. Hachem. Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing. *Comput Method Appl M*, 267:65 – 85, 2013.
- [51] Á. Csík. *Upwind Residual Distribution Schemes for General Hyperbolic Conservation Laws and Application to Ideal Magnetohydrodynamics*. PhD thesis, Katholieke Universiteit Leuven, Faculteit Wetenschappen Centrum voor PlasmaAstrofysica, Belgium, 2002.
- [52] Á. Csík and H. Deconinck. Space-time residual distribution schemes for hyperbolic conservation laws on unstructured linear finite elements. *Int J Numer Meth Fl*, 40(3-4):573–581, 2002.
- [53] Á. Csík, M. Ricchiuto, and H. Deconinck. A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws. *J Comput Phys*, 179(1):286 – 312, 2002.
- [54] 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(0):358 – 378, 2014.
- [55] Charles Dapogny and Pascal Frey. Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo*, 49(3):193–219, 2012.
- [56] D. de Santis. *Development of a high order residual distribution method for Navier Stokes and RANS equation*. PhD thesis, Univ. of Bordeaux I, 2013.

- [57] M.D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, and M. Napolitano. An immersed boundary method for compressible flows using local grid refinement. *J Comput Phys*, 225(2):2098 – 2117, 2007.
- [58] E. Van der Weide and H. Deconinck. Positive matrix distribution schemes for hyperbolic systems, with application to the euler equations. In *Computational fluid dynamics '96*, pages 747–753. John Wiley, Chichester, 1996.
- [59] Jean-Mathieu Deschenes, Guido S. Baruzzi, Patrick Lagace, and Wagdi G. Habashi. Fensap-ice: A cfd monte carlo approach to shed-ice trajectory and impact. In *SAE Technical Paper*. SAE International, 06 2011.
- [60] J. Dobeš. *Numerical Algorithms for the Computation of Steady and Unsteady Compressible Flow over Moving Geometries – Application to Fluid-Structure Interaction*. PhD thesis, Von Karman Institute for Fluid Dynamics, Universite Libre de Bruxelles, 2012.
- [61] J. Dobeš, Mario Ricchiuto, Rémi Abgrall, and Herman Deconinck. On hybrid residual distribution–galerkin discretizations for steady and time dependent viscous laminar flows. *Comput Method Appl M*, 283:1336 – 1356, 2015.
- [62] C. Dobrzynski. *Adaptation de Maillage anisotrope 3D et application à l'aéro-thermique des bâtiments*. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2005.
- [63] C. Dobrzynski and P. Frey. Anisotropic delaunay mesh adaptation for unsteady simulations. Proc.of 17th Int. Meshing Roundtable, Pittsburgh, USA, 2008.
- [64] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. *Encyclopedia of Computational Mechanics*, chapter 14, Arbitrary Lagrangian-Eulerian Methods. John Wiley & Sons, Ltd, 2004.
- [65] V. Ducrot and P. Frey. *Anisotropic Level Set Adaptation for Accurate Interface Capturing*, pages 159–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [66] H. Dütsch, F. Durst, S. Becker, and H. Lienhart. Low-reynolds-number flow around an oscillating circular cylinder at low keulegan–carpenter numbers. *J Fluid Mech*, 360:249–271, 4 1998.
- [67] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ale schemes

- for the solution of flow problems on moving grids. *J Comput Phys*, 174(2):669 – 694, 2001.
- [68] Charbel Farhat and Vinod K. Lakshminarayan. An {ALE} formulation of embedded boundary methods for tracking boundary layers in turbulent fluid–structure interaction problems. *J Comput Phys*, 263:53 – 70, 2014.
- [69] A. Ferrante and H. Deconinck. Solution of the unsteady euler equations using residual distribution and flux corrected transport. Technical report, Von Karman Institute for Fluid Dynamics, 1997.
- [70] P.J. Frey and F. Alauzet. Anisotropic mesh adaptation for {CFD} computations. *Comput Method Appl M*, 194(48–49):5068 – 5082, 2005. Unstructured Mesh Generation.
- [71] A. Froehly. *Couplage d’un schéma aux résidus distribués à l’analyse isogéométrique : méthode numérique et outils de génération et adaptation de maillage*. PhD thesis, Univ. of Bordeaux I, 2012.
- [72] T. J.R. Hughes G. Scovazzi. Lecture notes on continuum mechanics on arbitrary moving domains. Lecture Notes, November 2007.
- [73] R. Ghas, R. Mittal, and H. Dong. A sharp interface immersed boundary method for compressible viscous flows. *J Comput Phys*, 225(1):528 – 553, 2007.
- [74] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J Comput Phys*, 169(2):363 – 426, 2001.
- [75] H. Gopalan and A. Povitsky. Lift enhancement of flapping airfoils by generalized pitching motion. *J Aircraft*, 47(6):1884–1897, 2010.
- [76] G.J. Gorman, M.D. Piggott, C.C. Pain, C.R.E. de Oliveira, A.P. Umpleby, and A.J.H. Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Modelling*, 12(3–4):436 – 452, 2006.
- [77] Yannick Gorsse, Angelo Iollo, Haysam Telib, and Lisl Weynans. A simple second order cartesian scheme for compressible euler flows. *J Comput Phys*, 231(23):7780 – 7794, 2012.
- [78] A. Guardone, D. Isola, and G. Quaranta. Arbitrary lagrangian eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping. *J Comput Phys*, 230(20):7706 – 7722, 2011.

- [79] Hervé Guillard and Charbel Farhat. On the significance of the geometric conservation law for flow computations on moving meshes. *Comput Method Appl M*, 190(11–12):1467 – 1482, 2000.
- [80] S.M.J. Guzik and C.P.T. Groth. Comparison of solution accuracy of multidimensional residual distribution and godunov-type finite-volume methods. *Int J Comput Fluid D*, 22(1-2):61–83, 2008.
- [81] D. Guégan, O. Allain, A. Dervieux, and F. Alauzet. An  $l_\infty$ – $l_p$  mesh-adaptive method for computing unsteady bi-fluid flows. *Int J Numer Meth Eng*, 84(11):1376–1406, 2010.
- [82] E. Hachem, S. Feghali, R. Codina, and T. Coupez. Immersed stress method for fluid–structure interaction using anisotropic mesh adaptation. *Int J Numer Meth Eng*, 94(9):805–825, 2013.
- [83] J.-W. He, R. Glowinski, R. Metcalfe, A. Nordlander, and J. Periaux. Active control and drag optimization for flow past a circular cylinder. *J Comput Phys*, 163:83–117, 2000.
- [84] R.D. Henderson. Nonlinear dynamics and pattern formation in turbulent wake transition. *J Fluid Mech*, 352:65–112, 1997.
- [85] C.W Hirt, A.A Amsden, and J.L Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *J Comput Phys*, 14(3):227 – 253, 1974.
- [86] Matthew Hubbard. Non-oscillatory third order fluctuation splitting schemes for steady scalar conservation laws. *J Comput Phys*, 222(2):740 – 768, 2007.
- [87] M.E. Hubbard and M.J. Baines. Conservative multidimensional upwinding for the steady two-dimensional shallow water equations. *J Comput Phys*, 138(2):419 – 448, 1997.
- [88] M.E. Hubbard and A.L. Laird. Achieving high-order fluctuation splitting schemes by extending the stencil. *Comput Fluids*, 34(4–5):443 – 459, 2005. Residual Distribution Schemes, Discontinuous Galerkin Schemes and Adaptation.
- [89] M.E. Hubbard and M. Ricchiuto. Discontinuous upwind residual distribution: A route to unconditional positivity and high order accuracy. *Comput Fluids*, 46(1):263 – 269, 2011. 10th {ICFD} Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).

- [90] Thomas J R Hughes, Leopoldo P. Franca, and Gregory M. Hulbert. A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-diffusive equations. *Comput Method Appl M*, 73(2):173–189, 1989.
- [91] Thomas J. R. Hughes, Guglielmo Scovazzi, and Leopoldo P. Franca. *Multiscale and Stabilized Methods*. John Wiley & Sons, Ltd, 2004.
- [92] Thomas J. R. Hughes, Guglielmo Scovazzi, and Tayfun E. Tezduyar. Stabilized methods for compressible flows. *J Sci Comput*, 43(3):343–368, 2010.
- [93] Thomas J.R. Hughes, Guglielmo Scovazzi, Pavel B. Bochev, and Annalisa Buffa. A multiscale discontinuous galerkin method with the computational structure of a continuous galerkin method. *Comput Method Appl M*, 195(19–22):2761 – 2787, 2006.
- [94] G. Jannoun. *Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady CFD problems*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2014.
- [95] A.A. Johnson and T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Method Appl M*, 119(1–2):73 – 94, 1994.
- [96] A.A. Johnson and T.E. Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Comput Method Appl M*, 134(3–4):351 – 373, 1996.
- [97] Vinay Kalro and Tayfun E. Tezduyar. A parallel 3d computational method for fluid–structure interactions in parachute systems. *Comput Method Appl M*, 190(3–4):321 – 332, 2000.
- [98] Khodor Khadra, Philippe Angot, Sacha Parneix, and Jean-Paul Caltagirone. Fictitious domain approach for numerical modelling of navier–stokes equations. *Int J Numer Meth Fl*, 34(8):651–684, 2000.
- [99] Rooh. A. Khurram and Arif Masud. A multiscale/stabilized formulation of the incompressible navier–stokes equations for moving boundary flows and fluid–structure interaction. *Comput Mech*, 38(4):403–416, 2006.
- [100] T. Kilian. Experimental investigation of generic ica block trajectories under freestream conditions. In *German Aerospace Congress*, September 2016.
- [101] T. Kinsey and G. Dumas. Parametric study of an oscillating airfoil n a power-extraction regime. *AIAA Journal*, 46(6):543–561, 2008.

- [102] M.P. Kirkpatrick, S.W. Armfield, and J.H. Kent. A representation of curved boundaries for the solution of the navier–stokes equations on a staggered three-dimensional cartesian grid. *J Comput Phys*, 184(1):1 – 36, 2003.
- [103] David Kohlman and Robert Winn. *Analytical prediction of trajectories of ice pieces after release in an airstream*. Aerospace Sciences Meetings. AIAA Journal, Jan 2001.
- [104] Dmitry Kolomenskiy and Kai Schneider. A fourier spectral method for the navier–stokes equations with volume penalization for moving solid obstacles. *J Comput Phys*, 228(16):5687 – 5709, 2009.
- [105] L. Koloszár, N. Villedieu, T. Quintino, P. Rambaud, H. Deconinck, and J. Anthoine. Residual distribution method for aeroacoustics. *AIAA Journal*, 49(5):1021 – 1037, 2011.
- [106] A. Larat. *Conception and Analysis of Very High Order Residual Distribution Schemes. Application to Fluid Mechanics*. PhD thesis, Univ. of Bordeaux I, 2009.
- [107] Michel Lesoinne and Charbel Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput Method Appl M*, 134(1):71 – 90, 1996.
- [108] Randall J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, New York, 2002. Autres tirages : 2003, 2005, 2006.
- [109] C.-C. Liao, Y.-W. Chang, C.-A. Lin, and J.M. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Comput Fluids*, 39(1):152 – 167, 2010.
- [110] Qianlong Liu and Oleg V. Vasilyev. A brinkman penalization method for compressible flows in complex geometries. *J Comput Phys*, 227(2):946 – 966, 2007.
- [111] A. Madrane. *Parallel Implementation of a Dynamic Overset Unstructured Grid Approach*, pages 733–739. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [112] J. Maerz and G. Degrez. Improving time accuracy of residual distribution schemes. Technical report, Von Karman Institute for Fluid Dynamics, 1996.

## BIBLIOGRAPHY

---

- [113] Durbin P. Majumdar S., Iaccarino G. Rans solvers with adaptive structured boundary non-conforming grids, 2001.
- [114] Alireza Mazaheri and Hiroaki Nishikawa. Very efficient high-order hyperbolic schemes for time-dependent advection–diffusion problems: Third-, fourth-, and sixth-order. *Comput Fluids*, 102:131 – 147, 2014.
- [115] Alireza Mazaheri and Hiroaki Nishikawa. Improved second-order hyperbolic residual-distribution scheme and its extension to third-order on arbitrary triangular grids. *J Comput Phys*, 300:455 – 491, 2015.
- [116] M. Mezine. *Conception de Schmas Distributifs pour l’aérodynamique stationnaire et instationnaire*. PhD thesis, Université de Bordeaux I, 2002.
- [117] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu Rev Fluid Mech*, 37:239–261, 2005.
- [118] F. Morency, H. Beaugendre, and F. Gallizio. Aerodynamic force evaluation for ice shedding phenomenon using vortex in cell scheme, penalisation and level set approaches. *Int J Comput Fluid D*, 26:435–450, 2012.
- [119] Scott Morton, Robert Tomaro, and Ralph Noack. chapter An Overset Unstructured Grid Methodology Applied to a C-130 with a Cargo Pallet and Extraction Parachute. Aerospace Sciences Meetings. AIAA, Jan 2006.
- [120] P. Garcia Navarro, M.E. Hubbard, and A. Priestley. Genuinely multi-dimensional upwinding for the 2d shallow water equations. *J Comput Phys*, 121(1):79 – 93, 1995.
- [121] Guoxi Ni, Song Jiang, and Kun Xu. Remapping-free ale-type kinetic method for flow computations. *J Comput Phys*, 228(8):3154 – 3171, 2009.
- [122] Hiroaki Nishikawa and Philip Roe. *High-Order Fluctuation-Splitting Schemes for Advection-Diffusion Equations*, pages 77–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [123] L. Nouveau, R. Abgrall, H. Alcin, H. Beaugendre, and C. Dobrzynski. Residual distribution schemes for penalized navier stokes equations on adapted grids. In *ECCM V - ECFD VI*, Barcelona, Spain, July 2014.
- [124] L. Nouveau, H. Beaugendre, C. Dobrzynski, R. Abgrall, and M. Ricchiuto. An adaptive, residual based, splitting approach for the penalized navier stokes equations. *Comput Method Appl M*, 303:208 – 230, 2016.

- [125] L. Nouveau, H. Beaugendre, M. Ricchiuto, Cecile Dobrzynski, and R. Abgrall. An adaptive ALE residual based penalization approach for laminar flows with moving bodies. Research Report RR-8936, INRIA Bordeaux, équipe CARDAMOM, July 2016.
- [126] H. Paillère, J. Boxho, G. Degrez, and H. Deconinck. Multidimensional upwind residual distribution schemes for the convection-diffusion equation. *Int J Numer Meth Fl*, 23(9):923–936, 1996.
- [127] C.C. Pain, A.P. Umpheby, C.R.E. de Oliveira, and A.J.H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Comput Method Appl M*, 190(29–30):3771 – 3796, 2001.
- [128] P. De Palma, G. Pascazio, G. Rossiello, and M. Napolitano. A second-order-accurate monotone implicit fluctuation splitting scheme for unsteady problems. *J Comput Phys*, 208(1):1 – 33, 2005.
- [129] Michael Papadakis, Hsiung-Wei Yeong, and Ian Soares. *Simulation of Ice Shedding from a Business Jet Aircraft*. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, Jan 2007.
- [130] C.S. Peskin. Flow patterns around heart valves : A numerical method. *J Comput Phys*, 10:252–271, 1971.
- [131] P.W. Power, M.D. Piggott, F. Fang, G.J. Gorman, C.C. Pain, D.P. Marshall, A.J.H. Goddard, and I.M. Navon. Adjoint goal-based error norms for adaptive mesh ocean modelling. *Ocean Modelling*, 15(1–2):3 – 38, 2006. The Third International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows.
- [132] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-0479, Los Alamos Scientific Laboratory, 1973.
- [133] M. Ricchiuto. *Construction and analysis of compact residual discretizations for conservation laws on unstructured meshes*. PhD thesis, Von Karman Institute for Fluid Dynamics, 2005.
- [134] M. Ricchiuto. *Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows*. HDR Thesis, 2011.
- [135] M. Ricchiuto and R. Abgrall. Explicit runge–kutta residual distribution schemes for time dependent problems: Second order case. *J Comput Phys*, 229(16):5653 – 5691, 2010.

## BIBLIOGRAPHY

---

- [136] M. Ricchiuto, N. Villedieu, R. Abgrall, and H. Deconinck. On uniformly high-order accurate residual distribution schemes for advection–diffusion. *J Comput Appl Math*, 215(2):547 – 556, 2008. Proceedings of the Third International Conference on Advanced Computational Methods in Engineering (ACOMEN 2005). Proceedings of the Third International Conference on Advanced Computational Methods in Engineering (ACOMEN 2005).
- [137] Mario Ricchiuto. An explicit residual based approach for shallow water flows. *J Comput Phys*, 280:306 – 344, 2015.
- [138] Mario Ricchiuto and Andreas Bollermann. Stabilized residual distribution for shallow water simulations. *J Comput Phys*, 228(4):1071 – 1115, 2009.
- [139] Mario Ricchiuto, Árpád Csík, and Herman Deconinck. Residual distribution for general time-dependent conservation laws. *J Comput Phys*, 209(1):249 – 289, 2005.
- [140] P.L. Roe. Fluctuations and signals—a framework for numerical evolution problems. *Numer Meth Fl D*, pages 219 – 257, 1982.
- [141] A.M. Roma, C.S. Peskin, and Marsha J. Berger. An adaptative version of the immersed boundary method. *J Comput Phys*, 153:509–534, 1999.
- [142] C. J. Roy, C. C. Nelson, T. M. Smith, and C. C. Ober. Verification of euler/navier–stokes codes using the method of manufactured solutions. *Int J Numer Meth Fl*, 44(6):599–620, 2004.
- [143] E.M. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *J Comput Phys*, 123(2):450 – 465, 1996.
- [144] Luis de Castro Santos, Ramon Papa, and Marcelo Do Areal Ferrari. *A Simulation Model for Ice Impact Risk Evaluation*. Aerospace Sciences Meetings. AIAA, Jan 2003.
- [145] Niko Schade. chapter Simulation of Trajectories of Cuboid Cargos Released from a Generic Transport Aircraft. Fluid Dynamics and Co-located Conferences. AIAA, Jun 2011.
- [146] H. Schlichting. *Boundary-Layer Theory*. Mc-Graw-Hill, seventh edition edition, 1979.
- [147] K. Sermeus. *Multi-dimensional Upwind Discretization and Applications to Compressible Flows*. PhD thesis, Université libre de Bruxelles, VKI, 2013.

- [148] K. Shimoi. Numerical and experimental investigation of ice shedding, master thesis, whichita state university. Master Thesis, 2010.
- [149] Chi-Wang Shu. *High Order ENO and WENO Schemes for Computational Fluid Dynamics*, pages 439–582. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
- [150] Fotis Sotiropoulos and Xiaolei Yang. Immersed boundary methods for simulating fluid–structure interaction. *Prog Aerosp Sci*, 65:1 – 21, 2014.
- [151] KK Stein, TT Tezduyar, and RR Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *ASME J Appl Mech*, 70(1):58–63, 2003.
- [152] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM J Numer Anal*, 5(3):506–517, 1968.
- [153] R. Struijs and H. Deconinck. *Multidimensional upwind schemes for the Euler equations using fluctuation distribution on a grid consisting of triangles.*, pages 533–543. Vieweg+Teubner Verlag, Wiesbaden, 1990.
- [154] R. Struijs, H. Deconinck, P. De Palma, P.L. Roe, and K.G. Powell. *Progress on multidimensional upwind Euler solvers for unstructured grids.* AIAA 91–1550, June 1991.
- [155] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. An improved level set method for incompressible two-phase flows. *Comput Fluids*, 27(5–6):663 – 680, 1998.
- [156] H Tang and T Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J Numer Anal*, 41:487–515, 2003.
- [157] Anna-Karin Tornberg and Björn Engquist. A finite element based level-set method for multiphase flow applications. *Comput Visual Sci*, 3(1):93–101, 2000.
- [158] Yu-Heng Tseng and Joel H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *J Comput Phys*, 192(2):593 – 623, 2003.
- [159] H. S. Udaykumar, W. Shyy, and M. M. Rao. Elafint: A mixed eulerian–lagrangian method for fluid flows with complex and moving boundaries. *Int J Numer Meth Fl*, 22(8):691–712, 1996.
- [160] N. Villedieu. *High order discretisation by Residual Distribution schemes.* PhD thesis, Von Karman Institute for Fluid Dynamics, 2009.

## BIBLIOGRAPHY

---

- [161] N. Villedieu, T. Quintino, M. Ricchiuto, and H. Deconinck. Third order residual distribution schemes for the navier–stokes equations. *J Comput Phys*, 230(11):4301 – 4315, 2011. Special issue High Order Methods for {CFD} Problems.
- [162] H. Wang, R. Li, and T. Tang. Efficient computation of dendritic growth with r-adaptive finite element methods. *J Comput Phys*, 227(12):5984 – 6000, 2008.
- [163] T. Ye, R. Mittal, H.S. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J Comput Phys*, 156(2):209 – 240, 1999.
- [164] C.H. Zhou and J.Q. Ai. Mesh adaptation for simulation of unsteady flow with moving immersed boundaries. *Int J Numer Meth Fl*, 72(4):453–477, 2013.