T H È S E

présentée en vue de l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ BORDEAUX I

Spécialité : Mathématiques Appliquées

École Doctorale de Mathématiques et Informatique

Présentée et soutenue par

Jessica Hovnanian

****

Méthode de Frontières Immergées pour la Mécanique des Fluides. Application à la Simulation de la Nage.

****

Thèse dirigée par Angelo Iollo et Michel Bergmann

préparée à l'INRIA Bordeaux Sud Ouest, Equipe MC2

soutenue le lundi 17 Décembre 2012

- JURY -

Pierre Fabrie	-	Pr, Université Bordeaux I	- Président
Emmanuel Maitre	-	Pr, Université de Grenoble	- Rapporteur
Simone Camarri	-	CR, Université de Pise	- Rapporteur
Angelo Iollo	-	Pr, Université Bordeaux I	- Directeur de thèse
Michel Bergmann	-	CR, INRIA Bordeaux Sud Ouest	- Directeur de thèse
Cédric Galusinski	-	Pr, Université du Sud Toulon Var	- Examinateur

Résumé

Au cours de cette thèse, nous nous sommes intéressés à la modélisation des interactions fluide-structure entre un fluide visqueux, incompressible et une structure pouvant être déformable. Après avoir présenté les différentes approches possibles de modélisation, nous introduisons une nouvelle méthode de type frontière immergée : la méthode IPC ("Image Point Correction"). Combinant approches Ghost-Cell et Pénalisation, cette méthode mixte du second ordre globalement et localement en vitesse, est validée sur différents cas tests (comparaisons des coefficients aérodynamiques pour des cylindres fixes ou mobiles, sédimentation 2D d'un cylindre). Nous avons ensuite appliqué la méthode IPC à la simulation de la nage. Dans un premier temps, le solveur 2D a été couplé avec un algorithme d'optimisation mathématique afin de déterminer la loi de nage optimale pour une géométrie de poisson donnée. Puis, dans un second temps, nous avons simulé la nage 3D d'un poisson après reconstruction approchée de la géométrie basée sur des images du nageur. Enfin, grâce à l'outil du squelette, une reconstruction réaliste du poisson est proposée.

Mots clés : interactions fluide-structure, frontières immergées, Ghost-Cell, Pénalisation, simulation de la nage.

Immersed Boundary Method for the Fluid Mecanics applied to Fish-Like Swimming.

Abstract

The aim of this thesis is to investigate the modeling of fluid-structure interactions. The fluid is viscous and incompressible, and the structure is subject to an imposed deformation. After a survey of the different existing approachs to model fluid-structure interactions, we introduce a new immersed boundary method : the IPC method ("Image Point Correction"). This approach merges Ghost-Cell and Penalty concepts. It is globally and locally second order in velocity, and it is validated through several canonical simulations. Then, we apply the IPC method to fish-like swimming. First, the 2D solver is paired up with a mathematical optimization algorithm to determine the optimal swimming law for a given fish geometry. Secondly, we simulate a 3D swimmer after performing an approximated reconstruction of the geometry based on actual fish pictures. Finally, thanks to the skeleton approach, a realistic reconstruction of the fish is exposed.

Key words : fluid-structure interactions, immersed boundary, Ghost-Cell, Penalty method, fish-like swimming.

Équipe Projet INRIA MC2, INRIA Bordeaux Sud Ouest, 200 avenue de la Vieille Tour 33405, Talence Cedex.
Institut de Mathématiques de Bordeaux, UMR 5251, 351 cours de la libération, 33405, Talence Cedex.

Remerciements

A l'heure où cette thèse se termine, il est des gens qu'il est important de remercier car sans eux l'ensemble de ces travaux n'auraient pas pu voir le jour.

Les premières personnes que je souhaite remercier sont mes directeurs de thèse, Angelo IOLLO et Michel BERGMANN. Je remercie très vivement Angelo pour m'avoir fait confiance pendant ces trois années de thèse lorsque moi-même je ne me faisais plus confiance, pour m'avoir laissé m'entêter sur des problèmes qui n'auraient pas mériter autant de temps et surtout pour avoir toujours répondu présent malgré un emploi du temps très chargé.

Je souhaite également adresser tous mes remerciements à Michel. Michel a toujours fait preuve d'une patience et d'une écoute extraordinaires, a toujours répondu à toutes mes questions, même les plus ridicules et n'a jamais pester face à toutes mes erreurs d'étourderie. Je souhaite à tout doctorant d'avoir des directeurs aussi compréhensifs, patients, disponibles et à l'écoute. J'estime avoir eu énormément de chance lors de cette thèse.

Je remercie également très chaleureusement Emmanuel MAITRE et Simone CAMARRI qui m'ont fait l'honneur d'accepter de rapporter ce travail.

Je souhaite adresser toute ma reconnaissance à Cédric Galusinski pour avoir accepté de faire partie de ce jury ainsi que pour son aide et sa disponibilité lors de ma venue à Toulon malgré un emploi du temps chargé.

Je souhaite remercier particulièrement Pierre Fabrie qui, en plus d'avoir été le directeur de MATMECA pendant ma scolarité, m'a fait l'honneur d'accepter de faire partie de ce jury de thèse.

J'adresse tous mes remerciements à Christian Nguyen pour son aide pendant et après mon séjour à Toulon, pour sa bonne humeur et sa disponibilité.

Je tiens à remercier également Thierry Colin, Olivier Saut, Lisl Weynans, Clair Poignard, Iraj Mortazavi et tous les autres membres de l'équipe MC2. Leurs caractère et gentillesse uniques sont en grande partie une explication pour la bonne ambiance et générosité existantes au sein de l'équipe.

Je souhaite adresser tous mes remerciements à Anne-Laure Gautier, Pauline Raout, Catherine Vrit et Ida Sirben, pour leur aide et gentillesse récurrentes dans toutes les questions administratives me concernant au cours des trois dernières années.

Je tiens à remercier également Vincent, Thomas, Marco, Johana, Julie, Florian, Guillaume, Michael et Julien, (ex)doctorants et (ex)post-doctorants, toujours de bonne humeur, toujours gentils et souriants.

Je souhaite remercier particulièrement Yannick et Diane, qui auront été pour moi *la* rencontre pendant ces trois années de thèse. Les soirées remplies de rires auront été une constante avec vous et avoir eu Yannick comme cobureau pendant plus de trois ans fut un privilège.

Mes remerciements vont à tous mes amis Samy, Ious, Josepha, Grego, Vaness, Doro, Carole, Alain, Tom, Gael, Max, Armelle, Dewi, Cumcum, Clémence, Zouzou et tous les autres..., toutes ces personnes rencontrées grâce à MATMECA qui, en dépit de la distance nous séparant parfois, sont des proches. Mais des proches avec qui on souhaite toujours faire la fête !

Je souhaite adresser des remerciements spéciaux à des personnes formidables rencontrées pendant mes quelques années de pizzaïola. Je remercie sincèrement la seule et unique Dame Paquet qui s'occupe et se soucie toujours de nous, ses anciens et actuels employés, la Steph qui en plus d'avoir été ma collègue a aussi été ma colloque pendant les deux premières années de la thèse et qui a toujours été présente pour me relever quand ça n'allait pas en dépit de son emploi du temps de ministre ! Et enfin, je remercie Pascal, qui bien qu'il soit à des milliers de km maintenant, est toujours l'un de mes coups de coeur et n'a cessé de me pousser pour continuer.

Un grand merci à Alex, Jo et Jaou, qui sont mes amis depuis maintenant dix ans et plus, et qui ont toujours fait preuve d'une patience remarquable face à mon caractère... piquant ! Les soirées belottes et jeu de société ont été un véritable aparté durant la thèse.

Je souhaite adresser un immense merci à Jess et Emeline, ainsi que Etienne et Sylvain. Le moins que je puisse saluer est l'immense patience, gentillesse et compréhension dont vous avez tous toujours fait preuve et ces week-end à Limoges ont été une bouffée d'oxygène à chaque fois.

Je voudrais également remercier ma petite Mimi. Je ne sais s'il est des mots pour exprimer ma gratitude pour avoir une telle amitié. Une chose est sûre, t'avoir eu à mes côtés pendant ces trois dernières années était une nécessité. Merci d'avoir été là, même lorsque c'était à l'autre bout de la Terre.

Je souhaite remercier ma famille, et en particulier mes parents Philippe et Marina. Sans eux, je n'en serais pas là. Point de doute derrière cela.

Et enfin, the last but not the least, je souhaite remercier Quentin. Merci de me supporter chaque jour, d'endurer mon caractère sans pester, de me donner l'impression que je suis (presque) facile à vivre, et surtout d'avoir été à mes côtés dans les moments joyeux, comme les plus difficiles.

Table des matières

Introduction **1**

1 Préambule **7**
1.1 Les différents types de maillage . . . 8
1.2 Description Lagrangienne et Eulérienne . . . 10
1.2.1 Description lagrangienne . . . 11
1.2.2 Description eulérienne . . . 12
1.3 L'Outil Level Set . . . 13
1.3.1 Principe de la méthode . . . 13
1.3.2 Propriétés . . . 14

I Modélisation des Interactions Fluide - Structure **17**

2 Les Méthodes existantes d'interactions Fluide-Structure **19**
2.1 Les méthodes de frontières immergées . . . 20
2.1.1 Le "Continuous Forcing". . . . 22
2.1.2 Le "Discrete Forcing". . . . 25
2.2 La méthode ALE . . . 42
2.2.1 L'état de référence . . . 42
2.2.2 Formulation ALE des équations de Navier Stokes Incompressibles . 46
2.2.3 La régularisation de maillage . . . 47
2.2.4 L'adaptation de maillage . . . 48
2.2.5 Méthode ALE appliquée aux interactions fluide-structure . . . 48
2.3 Les méthodes de maillages adaptatifs . . . 49
2.3.1 L'AMR de Berger . . . 50
2.3.2 Les Octrees . . . 50
2.4 Conclusion . . . 51

3 La Méthode IPC (Image Point Correction) **53**
3.1 Résolution des équations de Navier Stokes incompressibles . . . 54
3.1.1 Les équations modélisant l'écoulement fluide . . . 54
3.1.2 Discrétisation des équations . . . 54
3.2 La méthode IPC . . . 57
3.2.1 Principe de la méthode . . . 58
3.2.2 Application aux équations de Navier Stokes . . . 61
3.2.3 Cas d'un solide en mouvement . . . 63
3.3 Résultats Numériques . . . 65
3.3.1 Étude de la convergence . . . 65
3.3.2 Cas d'un cylindre, fixe ou mobile . . . 66
3.3.3 Cas de deux cylindres fixes . . . 74

3.3.4 Sédimentation d'une bille 78
3.4 Conclusion 80

II Application à la Simulation de la Nage 83

4 Simulation Bidimensionnelle de la Nage 85
4.1 Définition du poisson 86
4.1.1 Les différents modes de nage 86
4.1.2 Les différentes catégories de poisson à nage ondulatoire 88
4.1.3 Définition du modèle géométrique du poisson 91
4.2 Optimisation 95
4.2.1 Éléments nécessaires à une bonne formulation d'un problème d'optimisation 95
4.2.2 Les méthodes de type déterministe 96
4.2.3 Les méthodes non déterministes 100
4.3 Optimisation de la nage 103
4.3.1 Optimisation des paramètres de nage 104
4.3.2 Trois paramètres de contrôle 113
4.3.3 Optimisation de la forme et de la nage du poisson 115
4.4 Conclusion 115

5 Simulation tridimensionnelle de la nage 119
5.1 L'outil Squelette 120
5.1.1 Définitions 120
5.1.2 La détermination numérique du Squelette 121
5.2 Simulation tridimensionnelle de la nage 124
5.2.1 Génération d'une géométrie tridimensionnelle approchée 124
5.2.2 Simulation 3D de la nage 131
5.3 Génération d'une géométrie tridimensionnelle réaliste 134
5.3.1 Redressement, reconstruction des fonctions Level Set et mise à l'échelle 136
5.3.2 Construction de la fonction Level Set 3D 137
5.4 Conclusion 146

Conclusion générale et perspectives 149

Bibliographie 153

Table des figures

1 Torsion du pont de Tacoma. 1
2 Effondrements d'éoliennes. 2
3 À gauche, le TGV japonais Shinkansen en haut et l'oiseau martin-pêcheur en bas ; à droite le modèle Bionic de Mercedes Benz en haut et le poisson coffre jaune en bas. 3

1.1 Éléments constitutifs d'un maillage 9
1.2 Maillages structurés 2D (en haut) et 3D (en bas) 9
1.3 Exemple de maillage homogène 2D 9
1.4 Exemple de maillage hétérogène 2D 10
1.5 Exemple de maillage non conforme 2D 10
1.6 Exemple de maillage hiérarchique 2D 10
1.7 Déformation d'un maillage lagrangien [69] 11
1.8 Évolution de deux interfaces circulaires croissantes : à gauche les deux interfaces sont distinctes ; à droite les interfaces ont fusionnées [115] 15
1.9 Approximation d'une interface avec des marqueurs : à gauche l'interface continue ; à droite l'interface discrète [115] 16

2.1 Différentes définitions de δ_h [71] 24
2.2 Schéma d'interpolation des vitesses aux bords du solide pour Fadlun *et al.* [37] 26
2.3 Représentation d'une frontière immergée, du ghost node G et des 2 points d'interpolation X_1 et X_2 utilisés par Tseng *et al.* pour imposer des conditions de bord [105]. 30
2.4 Traitement particulier appliqué à la méthode de Tseng *et al.* pour éviter les instabilités numériques [105] : (a) utilisation du point image I ; (b) modification locale de la frontière immergée. 30
2.5 Représentation d'une frontière immergée et des points utilisés par Mittal *et al.* pour leur méthode Ghost Cell [70]. GC désigne le noeud fantôme ("Ghost Cell"), BI désigne le point d'intersection à l'interface dans la direction normale ("Boundary Intersect") et IP représente le point image du noeud fantôme ("Image Point") 32
2.6 Représentation des différents cas de figure pouvant poser problème pour la détermination du point d'intersection à l'interface : (a) deux points d'intersection sont trouvés, (b) le point d'intersection n'est pas sur la frontière [70] 32
2.7 Schéma représentant la redéfinition des cellules fluides au voisinage de l'interface immergée pour la méthode Cut-Cell [122] 35
2.8 Schéma représentant l'ensemble des points utilisés pour la détermination du flux $f_{\mathtt{sw}}$ pour la méthode Cut-Cell [122] 35
2.9 Représentation d'une grille MAC 36
2.10 Représentation d'une cellule coupée par l'interface fluide-solide 38

2.11 Représentation d'une cellule coupée par l'interface fluide-solide 40
2.12 Grille associée à chaque description [35] 43
2.13 Comparaison de l'évolution des maillages lagrangien et ALE : (a) maillage éléments finis initial ; (b) maillage ALE à $t = 0.1$ms ; (c) maillage lagrangien à $t = 0.1$ms ; (d) zoom sur le maillage lagrangien [35] 43
2.14 Représentation des liens entre les différentes configurations [35] 44
2.15 Représentation des différents niveaux de hiérarchie dans le cas d'un AMR de Berger. L'échelle de gris représente la solution dans le cas d'une discontinuité [36]. 51
2.16 Représentation d'une cellule quad-tree. Les cellules blanches sont des cellules parents et les cellules grises sont des cellules enfants [126]. 51

3.1 Schématisation 2D de la localisation des différentes composantes des vitesses lors de la discrétisation spatiale. 55
3.2 Schéma 1D décrivant la méthodologie utilisée pour déterminer la vitesse des ghost cells 58
3.3 Schéma 1D représentant les deux types de méthode : (a) méthode IPC et (b) méthode de pénalisation classique, avec S le point solide soumis à la pénalisation, B le point sur l'interface et F le point fluide voisin. 59
3.4 Schéma 2D décrivant les éléments utilisés par le solveur pour déterminer la correction. La correction est appliquée au ghost points en utilisant les points symétriques. La vitesse du point symétrique est basée sur une interpolation utilisant les quatre voisins. 60
3.5 Représentation 3D de la position des points employés pour l'interpolation sur la vitesse du point symétrique 61
3.6 Distribution des iso-contours de pression dans le cas d'un cylindre fixe (à gauche) et d'un cylindre mobile (à droite). Les deux images correspondent à une solution pour une grille de taille (360×240). 63
3.7 (en haut, à gauche) : Répartition de l'erreur en u, (en haut à droite) : Répartition de l'erreur en v, (en bas) : Répartition de l'erreur en pression, toutes les images correspondent à la solution numérique pour la grille de taille (360×240). 66
3.8 Courbes d'ordre en norme L_2 (à gauche) et L_∞ (à droite) pour les composantes de vitesse u et v et pour la pression, en fonction de la taille de la grille. 67
3.9 Coefficient de trainée pour un cylindre fixe à $Re = 550$. 69
3.10 Coefficients de trainée et de portance à $Re = 200$ (en haut) et à $Re = 40$ (en bas). 69
3.11 Comparaison des coefficients de trainée pour un cylindre mobile à $Re = 550$, en utilisant la méthode IPC (en haut) et la pénalisation classique (en bas). 70
3.12 Comparaison des coefficients de trainée pour un cylindre mis en mouvement instantanément, à $Re = 40$ (en haut) et $Re = 200$ (en bas) , avec l'étude de Koumoutsakos *et al.* [61]. 71

3.13 Comparaison des coefficients de trainée pour un cylindre mis en mouvement, instantanément arrêté au temps $t = 5$ à $Re = 40$: notre méthode à gauche, celle de Koumoutsakos *et al.* à droite. 71
3.14 Isolignes de vorticité calculé avec notre méthode pour un cylindre mis en mouvement instantanément arrêté au temps $t = 5$ à $Re = 40$. 72
3.15 Isolignes de vorticité présenté par Koumoutsakos *et al.* dans [61], pour un cylindre mis en mouvement instantanément arrêté au temps $t = 5$ à $Re = 40$ 73
3.16 Comparaison des coefficients de trainée pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$: notre méthode à gauche, celle de Koumoutsakos *et al.* à droite [61]. 74
3.17 Isolignes de vorticité pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$. 75
3.18 Isolignes de vorticité pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$ présenté par Koumoutsakos *et al.* [61]. 76
3.19 Représentation des différents éléments géométriques pour la modélisation des cylindres en tandem. 77
3.20 Comparaison des coupes de vitesses verticales au niveau du cylindre au temps $t = 0.1$ entre notre étude et celle de Coquerelle et Cottet [30]. 79
3.21 Comparaison entre notre étude et celle de Coquerelle et Cottet [30] des évolutions en temps de la position du centre de la bille. 79
3.22 Comparaison entre notre étude et celle de Coquerelle et Cottet [30] des évolutions en temps de la vitesse verticale du centre de la bille. 80

4.1 Terminologie des différentes nageoires d'un poisson [94]. 87
4.2 L'anguille à gauche et la raie pastenague à droite. 87
4.3 Le baliste. 88
4.4 Evolution de la proportion du corps déformé lors de la nage de type BCF : (a) poisson anguilliforme, (b) poisson subcarangiforme, (c) poisson carangiforme et (d) poisson thunniforme [94]. 88
4.5 Les différentes catégories de poisson [94]. 89
4.6 L'anguille à gauche et la murène à droite. 89
4.7 La truite à gauche et le saumon à droite. 90
4.8 Le pompano à gauche et le requin marteau à droite. 90
4.9 Le thon rouge à gauche et l'espadon voilier à droite. 91
4.10 Transformation de Karman-Trefftz d'un cercle en un profil d'aile de longueur l : à gauche le cercle initial, à droite le profil d'aile obtenu après transformation [13]. 92
4.11 Géométries de poisson obtenues pour différentes valeurs de η : (a) $\eta = -0.004$, (b) $\eta = -0.003$, (c) $\eta = -0.002$ et (d) $\eta = -0.001$. 93
4.12 Simulation de la nage des quatre types de poisson au mode de nage BCF : (a) thunniforme, (b) carangiforme, (c) subcarangiforme et (d) anguilliforme. 95
4.13 Carte représentant les valeurs de $\mathcal{J}_1$ en fonction des paramètres c_1 et f. . . 107
4.14 Carte représentant les valeurs de $\mathcal{J}_2$ en fonction des paramètres c_1 et f. . . 110
4.15 Carte représentant les valeurs de $|\mathcal{J}_3|$ en fonction des paramètres c_1 et f. . 111

4.16 Évolution en temps de la vitesse horizontale d'un poisson aux paramètres $\eta_c = -0.04$, $\alpha = 5$, $l = 1$, $\lambda = 1.25$, $A = 0.15$, $c_1 = 1$ et $f = 3$. 112
4.17 Carte représentant les valeurs de $|\mathcal{J}_4|$ en fonction des paramètres c_1 et f. . 113

5.1 Définition du squelette d'un objet par la notion de boules maximales. . . . 121
5.2 Traitement d'une image doppler d'une carotide par le logiciel : à gauche l'image originale, à droite l'image après traitement [41]. 123
5.3 Délimitation d'une région initiale circulaire par l'utilisateur [41]. 123
5.4 Résultat final obtenu grâce au logiciel : la fonction Level Set et le squelette [41]. Les courbes de niveaux de gris représentent la fonction Level Set, et l'ensemble de points rouges, le squelette. 124
5.5 Représentation d'un chinchard : de haut en bas, profil puis dessus ; de gauche à droite, photo, contour et Level Set (ces deux derniers étant fournis par le logiciel). 126
5.6 Chinchard avant (à gauche) et après (à droite) redressement. 129
5.7 Représentation du chinchard reconstruit non déformé en haut, déformé en bas à gauche (vue de dessus) et avec maillage lagrangien en bas à droite. . 131
5.8 Partitionnements de domaine : inefficace à gauche, efficace à droite 132
5.9 Simulation de la nage du chinchard reconstruit 133
5.10 Comparaison des sillages 2D et 3D d'un poisson : en haut, sillage 2D, en bas, vue de dessus du sillage 3D. 134
5.11 Représentation d'un maquereau : de haut en bas, profil, dessus puis face ; de gauche à droite, photo, contour et Level Set (ces deux derniers étant fournis par le logiciel). 135
5.12 Vue de face d'un requin blanc : à gauche l'image originale, à droite l'image binaire produite par le logiciel. 136
5.13 Les trois vues du maquereau après redressement et mise à l'échelle : (a) vue de dessus, (b) vue de profil, (c) vue de face. Les axes rouges représentent les axes de symétrie. 137
5.14 Identification des différents cas pour la détermination de la Level Set 3D du poisson 138
5.15 Identification des différents paramètres sur la vue de face en vue des futures homothéties et translations. 140
5.16 Identification des hauteur et largeur réelles sur la vue de profil (en haut) et sur la vue de dessus redressée (en bas). 141
5.17 Définition des différents éléments pour la détermination de la fonction Level Set 3D dans le cas où le point considéré est en dehors du poisson (en haut) ou à l'intérieur du poisson (en bas). L'ensemble des points rouges représente le squelette médian redressé de la coupe transversale. 142
5.18 Définition de x_{curv} sur la vue de dessus déformée du poisson. L'ensemble de points rouges représente le squelette médian déformé de la vue de dessus. 144
5.19 Projection du point courant dans le plan orthogonal à la queue du poisson. $\mathbf{n}$ représente le vecteur normal au plan $\mathcal{P}$. 144
5.20 Reconstruction 3D du maquereau non déformé. 145
5.21 Reconstruction 3D du maquereau déformé. 146

Liste des tableaux

2.1 Tableau schématique comparant les différentes méthodes présentées. Les ordres sont donnés sur les vitesses uniquement et sont issus de [63] [30] [70] [122] [17]. Notons que l'ordre local de la méthode est directement lié à son habilité à gérer les simulations à nombre de Reynolds élevé à grilles égales. 41

3.1 Comparaison du coefficient de trainée avec des études précédentes. 70
3.2 Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 100$ et $L = 2.5D$ ou $L = 5.5D$. 77
3.3 Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 200$ et $L = 1.5D$ ou $L = 2D$. 77
3.4 Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 200$ et $L = 3D$ ou $L = 4D$. 78

4.1 Paramètres utilisés pour la forme et la nage pour les différents poissons. . . 94
4.2 Optimisation par la méthode du simplexe de la puissance moyenne dépensée par le poisson sur une distance L_0. 106
4.3 Optimisation par la méthode du simplexe de la puissance moyenne dépensée par le poisson sur une distance L_0 pour un second jeu de paramètres. . . . 107
4.4 Optimisation par la méthode du recuit-simulé de la puissance moyenne dépensée par le poisson sur une distance L_0. 108
4.5 Optimisation par la méthode du simplexe de l'énergie totale dépensée par le poisson sur une distance L_0. 109
4.6 Optimisation par la méthode du simplexe de l'index propulsif du poisson sur une distance L_0. 110
4.7 Optimisation par la méthode du simplexe de l'index propulsif du poisson en régime établi sur une distance $\tilde{L}_0$. 112
4.8 Optimisation par la méthode du simplexe des différentes fonctionnelles pour trois paramètres de contrôle de nage. 114
4.9 Optimisation par la méthode du simplexe des différentes fonctionnelles avec un paramètre de contrôle de forme et deux paramètres de contrôle de nage. 115

Introduction

La multitude des phénomènes physiques faisant interagir une structure et un fluide rend la connaissance de la modélisation des interactions fluide-structure très attractive. Par exemple, on retrouve ces phénomènes d'interactions lors de la modélisation de phénomènes tels que l'écoulement de l'air sur une aile d'avion ou sur les palles d'une éolienne, ou bien encore l'étude des flux marins sur la structure d'un bâteau, le mât d'une éolienne ou les piles d'un pont. Ces problèmes se retrouvent également dans les domaines de la biomécanique ou de la médecine, à travers l'étude de la nage des poissons ou des écoulements sanguins dans les vaisseaux.

L'importance de la connaissance des interactions fluide-structure s'est beaucoup accrue suite à des accidents qui sont aujourd'hui bien connus. On pense entre autre à l'effondrement du pont de Tacoma en 1940 dont la structure était entrée en résonance avec les tourbillons du vent, ce qui provoqua d'abord le tremblement de la structure, puis sa torsion pour terminer par son effondrement (voir Figure 1 [1]). Un autre évènement célèbre fut l'effondrement en 1850 du pont de la Basse Chaîne pendant la traversée d'une troupe militaire dont la fréquence de marche était entrée en résonance avec celle du pont.



FIGURE 1 – Torsion du pont de Tacoma.

L'importance de la connaissance des interactions fluide-structure se manifeste également chez les éoliennes, par exemple. En effet, des palles d'éoliennes peuvent se déformer suite à l'influence du vent sur ces dernières, heurtant alors son mât et conduisant à la destruction de l'éolienne (voir Figure 2 [2]). Tous ces accidents ont démontré l'importance que la connaissance des interactions fluide-structure peut présenter.

De plus, les performances technologiques dans le domaine sous-marin n'égalent toujours pas celles des animaux marins de nos jours. Ces espèces étant issues de milliers d'années d'évolution, leur forme et leur nage se sont optimisées au cours du temps afin d'être plus efficaces pour un objectif précis (exemple : chasse, fuite, camouflage...). Les ingénieurs cherchant à atteindre de telles performances en sont venus à développer un

1. source : zebac.free.fr/visiteur/ponts/pages/cont5.html
2. http ://www.popscreen.com/v/6WgXx/Destruction-dune-éolienne-par-le-vent



FIGURE 2 – Effondrements d'éoliennes.

domaine physique à part entière baptisé le *biomimétisme*. Au sein de cette branche, les chercheurs tentent de concevoir des engins en s'inspirant des espèces vivantes (par exemple : les revêtements autonettoyants pour toits, vitres ou peintures sont copiés sur les feuilles de lotus, la recherche sur la diminution des nuisances sonores des trains a conduit certains ingénieurs à s'inspirer du hibou, ou encore les japonais à s'intéresser à l'oiseau martin-pêcheur pour diminuer la résistance de l'air sur le train, ou enfin, le groupe Mercedes Benz s'est inspiré du poisson coffre jaune pour créer son modèle "Bionic" (voir Figure 3 [3])). Ce domaine est en pleine expansion aujourd'hui et a connu un véritable engouement en 1935 lors de l'énoncé du paradoxe de Gray [46]. James Gray stipulait que les muscles des dauphins devraient être sept fois plus puissants qu'ils ne l'étaient pour permettre aux dauphins de nager aux vitesses enregistrées. Bien que ce paradoxe trouva une réponse en 2008 grâce à Timothy Wei, il n'enlève en rien l'intérêt en plein essor pour la mécanique sous-marine.

Tous ces phénomènes multi-physiques font par ailleurs intervenir deux milieux distincts n'obéissant pas aux mêmes lois fondamentales, tout en interagissant l'un avec l'autre. En outre, les descriptions classiques pour décrire chaque milieu sont également différentes : le formalisme eulérien est usuellement utilisé pour décrire le fluide tandis que le solide est habituellement suivi de manière lagrangienne. La principale différence entre les deux descriptions réside dans le point de référence utilisé pour décrire les mouvements observés : la description lagrangienne se placera par rapport à une configuration initiale donnée tandis que le point de vue eulérien se situe dans la configuration déformée à chaque instant. Toutes ces différences de modélisation rendent la tâche du couplage fluide-structure particulièrement délicate et stimulante.

Afin de pouvoir modéliser ces phénomènes, nous nous plaçons sur une échelle intermédiaire de la physique, l'échelle *mésoscopique*, qui constitue un choix raisonnable

3. source : http ://www.ecolomagtunisie.com/energie/le-biomimetisme-des-solutions-durables-inspirees-de-la-nature.html/



FIGURE 3 – À gauche, le TGV japonais Shinkansen en haut et l'oiseau martin-pêcheur en bas ; à droite le modèle Bionic de Mercedes Benz en haut et le poisson coffre jaune en bas.

pour cette modélisation. Elle est située entre l'échelle macroscopique (qui décrit des phénomènes à l'échelle de l'homme) et l'échelle microscopique. Quand l'échelle microscopique représentent des éléments de l'ordre du nm, l'échelle mésoscopique, quant à elle, décrit des éléments de l'ordre du μm. Elle est très bien adaptée pour décrire des grandeurs physiques telles que la pression, la vitesse ou encore la masse volumique, sur un volume fluide. Elle permet de les décrire de manière continue contrairement à l'échelle microscopique qui décrit des phénomènes au niveau de l'atome et où cette hypothèse de continuité des variables ne peut plus être respectée.

Des avancées ont d'ores et déjà été effectuées avec l'introduction de la méthode de frontières immergées par Charles Peskin en 1972 [82], approche qui fut une base par la suite pour de nombreuses nouvelles méthodes de modélisation d'interactions fluide-structure sur maillage cartésien. Basée sur un concept complètement différent, la méthode ALE ("Arbitrary Lagrangian Eulerian") présentée dans les années 60, s'articule autour d'un maillage adapté au(x) solide(s) présent(s), combinant les descriptions lagrangienne et eulérienne. Enfin, une avancée importante fut l'introduction en 1983 par Marsha J. Berger des techniques d'adaptation de maillage.

La variété de ces techniques démontrent le grand intérêt que représente la modélisation des interactions fluide-structure.

Toutefois, aucune méthode n'est parfaite et aujourd'hui encore, la modélisation précise, robuste, rapide et simple d'implémentation de phénomènes multi-dimensionnels reste un objectif pour les ingénieurs.

Objectifs

C'est dans ce contexte d'interactions fluide-structure que se situent les travaux menés durant cette thèse. Nous avons cherché à modéliser des écoulements de fluides incompressibles en présence de structure(s) solide(s), déformable(s) ou rigide(s), fixe(s) ou en mouvement. Nous avons donc choisi de développer une méthode sur maillage cartésien de type frontières immergées d'ordre 2, globalement et localement, au moins en vitesse.

Nous souhaitons garder le caractère cartésien de la modélisation pour pouvoir être en mesure d'utiliser des schémas numériques simples et d'envisager des simulations de phénomènes tridimensionnels, sur des temps de calcul relativement courts et d'avoir la possibilité de changer de géométrie sans que cela ait une influence quelconque sur le solveur.

Nous avons choisi d'appliquer cette méthode à la biomécanique à travers la modélisation de nageurs de type poisson.

Nous nous sommes placés en deux et trois dimensions.

En 2D, nous nous baserons initialement sur une formulation entièrement analytique pour modéliser le nageur. Ensuite, nous chercherons à optimiser la géométrie et la loi de nage du poisson à partir de méthodes d'optimisation mathématiques pour une fonctionnelle donnée (par exemple, la minimisation de l'énergie dépensée par le poisson sur une distance d donnée et en ne descendant pas en dessous d'une vitesse minimale donnée).

Notre objectif en trois dimensions consistera à reconstruire numériquement une géométrie de poisson réaliste, puis à le faire nager. Cette reconstruction s'effectuera grâce à l'outil mathématique du squelette. Cette dernière étude sera le fruit d'une coopération entre l'INRIA Bordeaux Sud Ouest et l'IMATH de Toulon. Cette collaboration s'est effecutée à travers le projet ANR Carpeinter [4] dont l'un des objectifs est la modélisation de problèmes de la dynamique des fluides appliqués à l'environnement et à la biologie, grâce à l'utilisation d'outils tels que les grilles cartésiennes, les fonctions Level Set et la méthode de pénalisation.

4. www.math.u-bordeaux1.fr/CARPEINTER

Plan de la thèse

Cette thèse s'articule en deux parties.

Avant d'entrer dans le vif du sujet, nous introduirons des notions et outils qui seront constamment utilisés dans la suite et qu'il est préférable, de définir en premier lieu afin qu'ils soient clairs et qu'il n'y ait pas besoin de les définir de manière inopinée dans la suite du manuscrit. Une fois ces éléments en main, nous pouvons entrer dans la première partie de la thèse.

Cette première partie, découpée en deux chapitres, constitue l'étude des méthodes de frontières immergées pour la modélisation d'interactions fluide-structure. Dans le premier chapitre, nous présenterons les trois grandes classes de méthodes numériques existantes dans ce domaine :

- les méthodes de frontières immergées,
- la méthode ALE,
- les méthodes de maillages adaptatifs.

Nous détaillerons plus précisément les méthodes de frontières immergées en présentant les deux approches possibles : le "Direct Forcing" et le "Discrete Forcing", tout en décrivant quelques méthodes distinctes existantes en leur sein.

Dans le second chapitre de cette partie, nous présenterons notre nouvelle méthode de frontières immergées, combinant "Direct" et "Discrete Forcing". Nous verrons la précision d'ordre 2 de cette méthode et la validerons en comparant ses résultats à ceux issus de la littérature, pour divers cas tests, pour des obstacles fixes et mobiles.

Dans la seconde partie de la thèse, également scindée en deux chapitres, nous appliquerons notre méthode de frontières immergées à la simulation de la nage d'un poisson.

Le premier chapitre de cette seconde partie se concentrera sur le cas bidimensionnel. Après avoir présenté les différents modes de nage et les diverses classes de poisson, nous optimiserons la forme et la nage d'un poisson 2D dont la géométrie et la loi de nage sont initialement analytiques. Nous verrons quels sont les profils et lois de nage optimisés pour des fonctions objectifs données.

Enfin, dans le deuxième chapitre de cette seconde partie, nous nous placerons dans le cas tridimensionnel. Nous chercherons ici à reconstituer une géométrie réaliste de poisson grâce à l'outil mathématique du squelette. Enfin, des simulations de nage seront effectuées pour le cas d'un poisson de type chinchard (ou "maquereau batard").

Chapitre 1

Préambule

Sommaire

1.1 Les différents types de maillage 8
1.2 Description Lagrangienne et Eulérienne 10
1.2.1 Description lagrangienne 11
1.2.2 Description eulérienne 12
1.3 L'Outil Level Set 13
1.3.1 Principe de la méthode 13
1.3.2 Propriétés 14

Le but de ce chapitre est de présenter des bases et des éléments qui seront fréquemment utilisés dans la suite de ce manuscrit. Ces différents éléments pourront paraître très décousus les uns des autres mais en réalité, tel un bricoleur qui crée sa boîte à outils en partant du mètre et allant jusqu'à la perceuse électrique, nous introduirons différents outils de travail, partant du plus simple, allant en se complexifiant, pour par la suite, les utiliser dans notre étude et ainsi mieux comprendre pourquoi nous avons choisi d'utiliser tel outil plutôt qu'un autre.

Ainsi, nous présenterons d'abord la notion de maillage, qui constitue une base pour un grand nombre de méthodes numériques. Puis, nous introduirons les notions de descriptions lagrangienne et eulérienne, qui sont les deux formalismes descriptifs utilisés en mécanique. Enfin, nous poserons l'outil Level Set. Conçue il y a une vingtaine d'années, la méthode Level Set est devenue une méthode très fréquemment utilisée pour la capture d'interface, employée dans de nombreux travaux [17] [43] [25] [70] [115] [24].

Lorsque l'on cherche une définition dans le dictionnaire pour "simuler", on trouve : *reproduire artificiellement une situation réelle à des fins de démonstration ou d'explication*[1]. C'est cet aspect que l'on retrouve dans la simulation numérique de phénomènes physiques, qu'il s'agisse par exemple de simuler des collisions de galaxies, un amerrissage, des phénomènes climatiques, des explosions nucléaires ou la nage des grands mammifères. Ces exemples illustrent également le rôle de la simulation numérique : il s'agit de reproduire des expériences qui sont infaisables par les physiciens (ex : une collision de planète)

1. définition trouvée sur le site internet deu Centre National de Ressources Textuelles et Lexicales : *www.cnrtl.fr*

ou dont le coût serait extrêmement élevé, spécialement si on le compare au coût de la simulation (ex : simuler un amerrissage).

Mais la simulation numérique n'est qu'une étape dans toute la méthodologie de simulation. Le processus complet se décline en quatre étapes : la théorie, la modélisation, la simulation et la vérification. Une fois la théorie définie derrière l'observation d'un phénomène physique, vient la définition du modèle. Ce modèle traduit la physique sous-jacente au travers d'équations mathématiques. Dans les cas simples, les équations se résolvent grâce à l'existence de solutions analytiques. Toutefois, la nature étant généralement loin d'être simple, on se retrouve la plupart du temps confronté à des équations dont on ne sait pas déterminer une solution analytique. Dès lors, deux étapes intermédiaires peuvent intervenir suivant la complexité du modèle créé : la réduction de modèle et l'expertise. La réduction de modèle peut s'effectuer soit par des méthodes asymptotiques [18] qui consistent à simplifier les équations par invariances, ordres de grandeur ou linéarisation par exemple, soit par des méthodes de projection qui réduisent l'espace de recherche (ex : la méthode POD [12]). Ensuite, l'expertise vérifie la validité du nouveau modèle obtenu et identifie la nouvelle gamme de paramètres admissibles s'il y a lieu. Une fois cette étape achevée, on utilise alors l'outil informatique pour simuler numériquement le phénomène. Simuler revient donc à déterminer l'évolution spatio-temporelle de variables physiques (vitesse, pression, température...). Néanmoins, lorsque l'on se place dans l'hypothèse du domaine continu, on considère que le domaine est constitué d'une infinité de points. Simuler équivaut alors à déterminer une infinité de valeurs, ce qui est impossible lorsque l'on ne dispose pas d'une solution analytique. On approxime alors le domaine d'étude en lui attribuant un nombre fini de points et on cherchera à déterminer les valeurs de nos variables en ces points. On associera à chacun de ces points physiques un volume élémentaire dont l'union reconstituera complètement notre domaine. Cet ensemble de points est ce que l'on appellera un *maillage*.

Les maillages peuvent différer de l'un à l'autre. C'est pourquoi nous définirons dans un premier temps différents types de maillage existants afin de mieux comprendre par la suite dans quelle configuration nous avons choisi de placer notre étude.

1.1 Les différents types de maillage

Définition 1.1 : un maillage

On appelle *maillage* une partition d'un domaine donné par un ensemble de sous-domaines. Ces sous-domaines sont appelés cellules (ou *mailles*). Les mailles sont formées par un ensemble de *sommets* (ou *noeuds*) reliés entre eux par des *arêtes* (voir Figure 1.1). L'ensemble des mailles forme ainsi le *maillage*.

Définition 1.2 : un maillage structuré

On appelle *maillage structuré* un maillage dont l'agencement des cellules est le même et répétitif (voir Figure 1.2[2]).

2. Ces images sont issues du cours de Jean Michel DISHCLER sur la Modélisation Géométrique

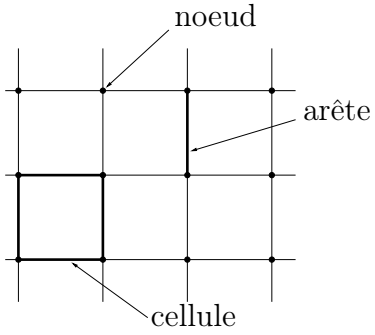


FIGURE 1.1 – Éléments constitutifs d'un maillage

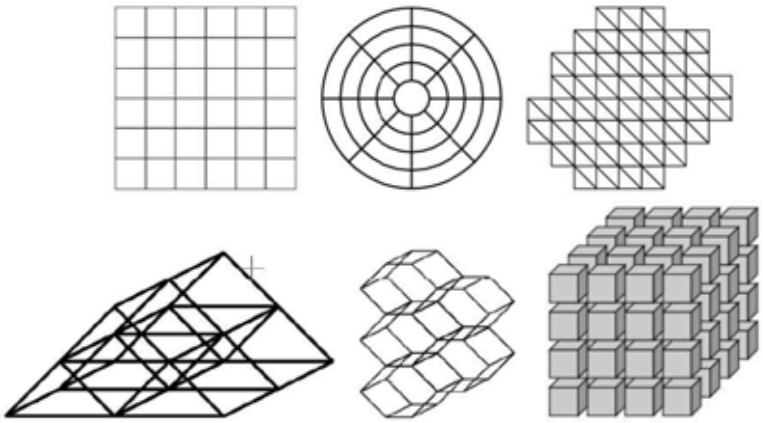


FIGURE 1.2 – Maillages structurés 2D (en haut) et 3D (en bas)

Définition 1.3 : un maillage homogène ou hétérogène

On appelle *maillage homogène* un maillage dont les cellules ont toujours le même nombre d'arêtes (voir Figure 1.3[3]). *A contrario*, un *maillage hétérogène* est un maillage qui n'est pas homogène (voir Figure 1.4[3]).

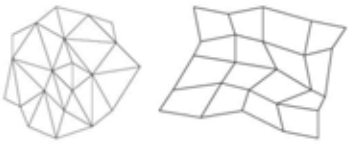


FIGURE 1.3 – Exemple de maillage homogène 2D

Définition 1.4 : un maillage conforme

On appelle *maillage conforme* un maillage dont aucun sommet n'appartient à l'arête d'une cellule (voir Figure 1.5[3]).

3. Ces images sont issues du cours de Jean Michel DISHCLER sur la Modélisation Géométrique



FIGURE 1.4 – Exemple de maillage hétérogène 2D



FIGURE 1.5 – Exemple de maillage non conforme 2D

Définition 1.5 : un maillage hiérarchique

On appelle *maillage hiérarchique* un maillage dont certaines cellules sont subdivisées en d'autres cellules (voir Figure 1.6[3]). Un tel maillage est généralement non conforme.

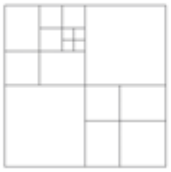


FIGURE 1.6 – Exemple de maillage hiérarchique 2D

Exemple de maillages :
- le maillage cartésien est un maillage structuré, homogène et conforme dont les mailles sont rectangulaires en 2D (voir Figure 1.1).
- les quad-trees en 2D (octrees en 3D) constituent un maillage structuré, homogène et non conforme, hiérarchique dont les mailles sont rectangulaires (voir Figure 1.6).

1.2 Description Lagrangienne et Eulérienne

Afin de pouvoir décrire précisément un objet ou un domaine, il est nécessaire de pouvoir le repérer au cours du temps. C'est dans ce but qu'on introduit la notion préalable de *référentiel*.

Le référentiel est une notion liée à l'observateur. Il représente l'ensemble des points animés du mouvement de l'observateur. À un référentiel, on associe un repère composé

d'un point d'origine et d'une base vectorielle. C'est le repère qui permet de matérialiser le référentiel et de repérer l'objet ou le domaine que l'on souhaite décrire.

Il existe deux types de description pour suivre un milieu : le formalisme eulérien et le formalisme lagrangien.

1.2.1 Description lagrangienne

Cette description est généralement privilégiée pour décrire les milieux matériels solides. Dans ce manuscrit, on se place dans le cas des milieux continus. On considère qu'un milieu continu est constitué d'un ensemble de particules. Dès lors, décrire le mouvement du milieu revient à déterminer les trajectoires de toutes les particules le constituant. On cherche donc à suivre ces particules au cours du temps.

Considérons un repère orthonormé fixe R associé à un référentiel $\mathcal{R}$. Considérons également un milieu continu de volume Ω_0 au temps de référence t_0, se déformant sous l'application d'une fonction φ.

On note $\Omega(t)$ la déformation de Ω_0 sous l'effet de φ, i.e. $\Omega(t) = \varphi(\Omega_0, t)$.

Plaçons nous maintenant au niveau des particules constituant le milieu. Si on note M_0 une particule matérielle repérée par sa position

$$\mathbf{X}(M_0, t = t_0) = \overrightarrow{OM_0},$$

alors la nouvelle position de cette particule à l'instant t est définie par :

$$\mathbf{x} = \varphi(\mathbf{X}, t).$$

L'application φ est donc déterminée par :

$$\begin{aligned} \varphi(., t) : \quad & \Omega_0 \to \Omega(t) \\ & \mathbf{X} \mapsto \mathbf{x} = \varphi(\mathbf{X}, t). \end{aligned}$$

On parvient ainsi à suivre les trajectoires des particules matérielles et φ représente la description lagrangienne du mouvement des particules dans le référentiel $\mathcal{R}$.

Afin que le problème soit correctement posé, l'application φ doit être bijective et régulière. On définit alors *le champ lagrangien des vitesses* comme le champ de vitesse des particules lagrangiennes.

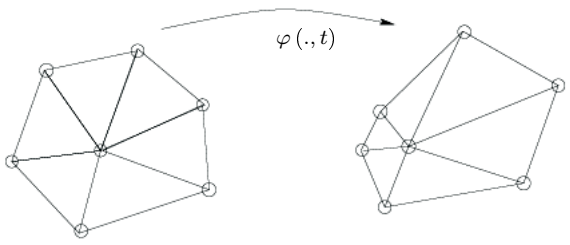


FIGURE 1.7 – Déformation d'un maillage lagrangien [69]

Soient u et γ la vitesse et l'accélération respectivement d'une particule se trouvant en $\mathbf{x} = \varphi(\mathbf{X}, t)$ à l'instant t. On a alors

$$\begin{cases} \mathbf{u}(\mathbf{x}, t) = \dfrac{\mathrm{d}\varphi}{\mathrm{d}t}(\mathbf{X}, t), \\ \\ \gamma(\mathbf{x}, t) = \dfrac{\mathrm{d}^2\varphi}{\mathrm{d}^2 t}(\mathbf{X}, t), \end{cases} \tag{1.1}$$

où $\dfrac{\mathrm{d}.}{\mathrm{d}t} = \left.\dfrac{\partial .}{\partial t}\right|_{\mathbf{x}}$.

La dérivée temporelle intervenant dans ces deux dernières équations est une dérivée dite *matérielle*, *particulaire* ou encore *lagrangienne*. La dérivée particulaire est souvent également notée avec un point au dessus de la quantité à dériver. Par conséquent, les équations (1.1) peuvent être réécrites comme

$$\begin{cases} \mathbf{u}(\mathbf{x}, t) = \dot{\varphi}(\mathbf{X}, t), \\ \\ \gamma(\mathbf{x}, t) = \ddot{\varphi}(\mathbf{X}, t). \end{cases}$$

Si on se place maintenant d'un point de vue numérique, les particules constituant le milieu d'étude sont positionnées aux noeuds du maillage. Puis leur nouvelle position est déterminée en appliquant la transformation φ.

Bien que cette approche soit généralement appliquée pour la modélisation des domaines solides, elle est également utilisée pour simuler des déplacements fluides [81] [60] [31]. Cette méthode fluide est appelée "méthode particulaire" [117] [87]. Elle a la capacité de pouvoir simuler des écoulements présentant des grandes déformations mais souffre d'une étape supplémentaire dans la modélisation : le remaillage. En effet, dans le suivi lagrangien des particules, celles-ci sont portées par l'écoulement et par conséquent se concentrent dans les zones aux gradients élevés. C'est pour éviter une trop forte accumulation des particules qu'on les remaille en créant de nouvelles particules, les répartissant sur les différents points de grille.

1.2.2 Description eulérienne

Il existe une autre description, dite eulérienne, généralement privilégiée pour décrire les mouvements fluides. Contrairement à la description lagrangienne qui caractérise son domaine déformé par rapport à sa configuration initiale et qui suit les particules, la description eulérienne se place en un point fixe et décrit l'évolution des particules qui passent en ce point. Par exemple, cela revient à se placer sur une bande d'arrêt d'urgence sur l'autoroute et regarder les voitures passer devant soi, tandis que la description lagrangienne se serait placée dans une voiture au départ et aurait déterminé sa trajectoire par rapport à son point de départ. La déscritpion eulérienne décrit donc le mouvement du domaine directement dans sa configuration déformée. On parle dans cette description de *champ eulérien des vecteurs vitesses*.

La dérivée temporelle du champ de déformation φ s'écrit du point de vue eulérien comme $\dfrac{\partial \varphi}{\partial t}$ où $\dfrac{\partial .}{\partial t} = \left.\dfrac{\partial .}{\partial t}\right|_{\mathbf{x}}$.

Numériquement, les choses sont extrêmement simples : après création d'un maillage fixe en temps, on considère que les noeuds du maillage contiennent les inconnues en vitesse (et autres inconnues) et on détermine à chaque itération en temps la valeur de nos inconnues en résolvant les équations mises en jeu. L'avantage de cette méthode est de pouvoir utiliser des schémas numériques simples (volumes finis, différences finies...) si on choisit d'utiliser un maillage cartésien.

Comme il l'a été mentionné, cette approche est généralement préférée pour décrire l'évolution d'un système fluide. Toutefois, tout comme la description lagrangienne est adaptable à un domaine fluide, la description eulérienne est également adaptable au suivi d'un système solide [69].

Les deux descriptions sont équivalentes. Il convient maintenant de présenter la relation liant les deux formulations.

Soient $\psi_1(\mathbf{X}, t)$ un champ de déformation lagrangien, et $\psi_2(\mathbf{x}, t)$ le même champ de déformation décrit eulériennement. Ces deux fonctions sont liées par la relation :

$$\begin{aligned} \psi_1(\mathbf{X}, t) &= \psi_2(\mathbf{x}, t) \\ &= \psi_2(\varphi(\mathbf{X}), t). \end{aligned}$$

Au niveau de la dérivation, on a :

$$\begin{aligned} \frac{\mathrm{d}\psi_1}{\mathrm{d}t}(\mathbf{X}, t) &= \frac{\partial \psi_2}{\partial t}(\mathbf{x}, t) + \frac{\partial \varphi}{\partial t}(\mathbf{X}, t) \cdot \nabla \psi_2(\mathbf{x}, t) \\ &= \frac{\partial \psi_2}{\partial t}(\mathbf{x}, t) + \mathbf{u} \cdot \nabla \psi_2(\mathbf{x}, t). \end{aligned}$$

1.3 L'Outil Level Set

Présentée pour la première fois en 1988 par Osher et Sethian [80], la méthode Level Set est devenue une approche de capture d'interface utilisée dans de nombreux domaines (traitement d'images, mécanique, propagation d'ondes etc..) par les numériciens en raison de ses propriétés avantageuses qui seront développées par la suite.

1.3.1 Principe de la méthode

La méthode Level Set consiste à repérer une interface Γ entre deux milieux par l'introduction de courbes de niveaux. Par conséquent, considérons deux domaines Ω_1 et Ω_2 séparés par l'interface Γ. On cherche à suivre le déplacement de Γ au cours du temps.

On introduit une fonction régulière ϕ dont Γ sera l'une des courbes de niveau. On choisit l'isoligne zero. Autrement dit :

$$\Gamma = \left\{ \mathbf{x} \in \mathbb{R}^2, \phi\left(\mathbf{x}\right) = 0 \right\}.$$

La définition de distance signée par rapport à l'interface est généralement utilisée pour définir la fonction Level Set en raison de ses propriétés particulièrement intéressantes qui seront développées par la suite. La fonction ϕ est alors définie par :

$$\phi\left(\mathbf{x}\right) = \begin{cases} -d\left(\mathbf{x}, \Gamma\right) & \text{si } \mathbf{x} \in \Omega_1, \\ +d\left(\mathbf{x}, \Gamma\right) & \text{si } \mathbf{x} \in \Omega_2, \end{cases}$$

où la fonction $d\left(\mathbf{x}, \mathbf{y}\right)$ désigne la fonction distance de $\mathbf{x}$ à $\mathbf{y}$.

On a choisi ici $\phi > 0$ dans Ω_2 et $\phi < 0$ dans Ω_1. Ce choix de convention revient à l'utilisateur.

Prenons par exemple le cas d'une interface circulaire 2D de rayon 1, centrée en $(0, 0)$. La fonction Level Set ϕ s'écrit alors :

$$\phi\left(\mathbf{x}\right) = \begin{cases} \sqrt{x^2 + y^2} - 1 & \text{si } \mathbf{x} \in \Omega_1, \\ -\left(\sqrt{x^2 + y^2} - 1\right) & \text{si } \mathbf{x} \in \Omega_2. \end{cases}$$

L'interface étant donc repérée par la courbe de niveau zéro de ϕ, capturer le déplacement de Γ revient à capturer l'isoligne 0 de ϕ. Si on considère que l'interface se déplace à la vitesse $\mathbf{u}$, alors son déplacement se traduit par l'équation de transport :

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{1.2}$$

La méthode Level Set permet de surcroît de gérer les changements topologiques de manière très naturelle.

Par exemple, si on considère deux interfaces circulaires initialement distinctes, croissantes au cours du temps, les deux cercles croissent jusqu'à fusion. L'équation de transport seule suffit pour que la fusion s'effectue (voir Figure 1.8).

1.3.2 Propriétés

La première propriété intéressante de la fonction Level Set est le fait qu'elle soit solution de l'équation eikonale :

$$|\nabla \phi| = 1. \tag{1.3}$$

Dans la mécanique des fluides, cette équation permet de redéfinir les courbes de niveau. En effet, si on déplace une interface Γ dans un milieu fluide avec l'équation (1.2) où $\mathbf{u}$ représente le champ de vitesse fluide, en supposant que l'on se place dans le cas général où la vitesse fluide n'est pas constante, alors les courbes de niveau ne sont pas déplacées avec la même vitesse partout dans le domaine d'étude. Par conséquent, le caractère de distance singée ne sera pas conservé par la fonction Level Set. D'où la

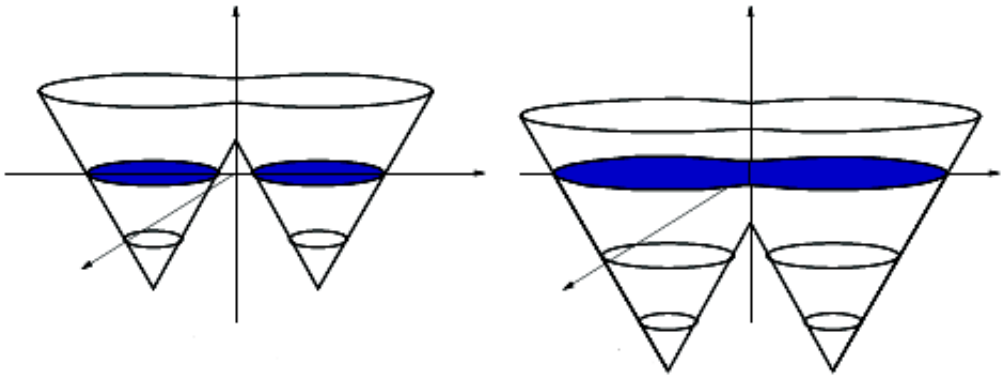


FIGURE 1.8 – Évolution de deux interfaces circulaires croissantes : à gauche les deux interfaces sont distinctes ; à droite les interfaces ont fusionnées [115]

nécessité d'une redistanciation des courbes de niveau en résolvant l'équation eikonale. Cette redistanciation s'effectue par rapport à l'isoligne zéro qui reste inchangée.

Les deux propriétés particulièrement intéressantes que l'on peut également obtenir à partir de la fonction Level Set due à sa définition en tant que distance signée sont :

1. la normale unitaire extérieure (intérieure lors du choix de la seconde convention de signe pour la fonction ϕ) à l'interface :
$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|_{\phi=0}},$$
2. la courbure moyenne à l'interface :
$$\begin{aligned} \kappa &= \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right)_{\phi=0} \\ &= \nabla \cdot \mathbf{n}. \end{aligned}$$

Dans le cas particulier où une formule analytique est connue pour la fonction Level Set, $\mathbf{n}$ et κ peuvent être déterminés directement grâce à l'outil Level Set. Dans le cas général où l'on ne possède pas de formule analytique pour ϕ, on doit faire appel au numérique. L'approche numérique que nous avons choisi dans nos travaux est la suivante : on approxime l'interface Γ en considérant qu'elle est constituée d'un certain nombre de marqueurs lagrangiens dépourvus de masse (voir Figure 1.9). Dès lors, la distance d'un noeud de grille à l'interface Γ est approximée par la plus petite distance du noeud de grille à chaque marqueur lagrangien. Notons que l'on aurait également pu choisir par exemple de déterminer la distance projetée de chaque noeud du maillage sur l'interface sur l'interface.

Ainsi, au travers de ce chapitre, nous avons présenté les différents types de maillage, les formalismes lagrangien et eulérien, et les principe et propriétés de la méthode Level Set. Dans les chapitres suivants nous verrons combien ces notions et outils sont utilisés fréquemment dans les problèmes d'interactions fluide-strucutre.

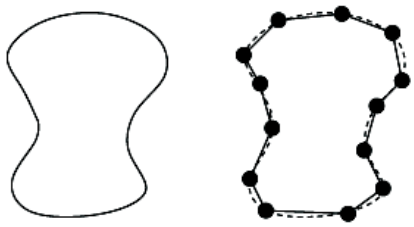


FIGURE 1.9 – Approximation d'une interface avec des marqueurs : à gauche l'interface continue ; à droite l'interface discrète [115]

Première partie

Modélisation des Interactions Fluide - Structure

Chapitre 2

Les Méthodes existantes d'interactions Fluide-Structure

Sommaire

2.1 Les méthodes de frontières immergées 20
2.1.1 Le "Continuous Forcing". 22
2.1.2 Le "Discrete Forcing". 25
2.2 La méthode ALE 42
2.2.1 L'état de référence 42
2.2.2 Formulation ALE des équations de Navier Stokes Incompressibles . . 46
2.2.3 La régularisation de maillage 47
2.2.4 L'adaptation de maillage 48
2.2.5 Méthode ALE appliquée aux interactions fluide-structure 48
2.3 Les méthodes de maillages adaptatifs 49
2.3.1 L'AMR de Berger 50
2.3.2 Les Octrees 50
2.4 Conclusion 51

À travers ce chapitre, notre objectif est de présenter trois approches distinctes pour la modélisation d'interactions fluide-structure. Le premier point de vue est de considérer un maillage fixe, indépendant de la structure présente. Cette approche permet d'utiliser des schémas numériques simples, de gérer les déplacements du solide et d'envisager des configurations 3D sans difficultés *a priori*. L'écueil se situe dans la précision des schémas. En effet, l'interface fluide-structure ne coïncidant, *a priori*, pas avec les noeuds du maillage, l'imposition précise de conditions de bords devient alors une tâche délicate. Les méthodes de frontières immergées constituent cette première classe de méthodes et se distinguent les unes des autres à travers leur manière d'imposer les conditions à l'interface.

Le point de vue opposé est l'approche ALE qui définit un maillage adapté à la structure immergée. Cette méthode est très précise mais devient très coûteuse dès que l'on considère un solide en mouvement ou que l'on souhaite se placer en 3D.

Enfin nous avons les méthodes de maillages adaptatifs. Ces méthodes se basent également sur un maillage de base fixe et indépendant du solide. Toutefois ce maillage peut se raffiner localement près de l'interface fluide-structure ou dans les zones présentant des phénomènes plus fins, augmentant ainsi la précision. Simple dans l'idée, cette méthode permet de gérer les déplacements de structure et le passage au 3D, mais exige un raffinement différent à chaque instant, dépendant de la position de l'interface, et des schémas numériques adaptés dès que l'on déplace l'interface.

2.1 Les méthodes de frontières immergées

Le terme de "méthodes de frontières immergées" fut utilisé pour la première fois pour qualifier une méthode développée par Peskin en 1972 cherchant à simuler les mécanismes cardiaques et ses flux sanguins [82]. L'aspect novateur de cette méthode était dû au fait que l'entière simulation était effectuée sur une grille cartésienne eulérienne non adaptée à la géométrie du coeur et consistait en la résolution d'un unique système d'équations ; le fluide mis en jeu était suivi de manière totalement eulérienne tandis que le solide était décrit suivant une configuration lagrangienne. De plus, une approche nouvelle était introduite pour imposer les effets de la frontière immergée sur l'écoulement. Il existe également une autre classe de méthodes, habituellement référencées comme "les méthodes de grilles cartésiennes " (ou méthodes Cut-Cell) qui ont été a l'origine développées pour simuler des écoulements non visqueux en présence de frontières solides, complexes et immergées sur des grilles cartésiennes [11], [29], [123]. Ces méthodes ont été étendues pour simuler des écoulements visqueux instables [111], [122]. Elles ont des capacités similaires à celles des méthodes de frontières immergées. Dans notre approche, nous utiliserons le terme de méthode de frontières immergées pour inclure toutes les méthodes qui simulent des écoulements visqueux avec des frontières immergées sur des grilles non adaptées au solide. En outre, nous ne nous concentrerons que sur des méthodes de frontières immergées avec des frontières solides. Toutefois, les méthodes de frontières immergées ont également été étendues pour modéliser des interfaces liquide/liquide ou liquide/gaz [98], [1], [92], [104] mais également pour simuler des phénomènes tels que la solidification [121].

La méthode de frontières immergées présentée par Peskin possède néanmoins plusieurs inconvénients. Tout d'abord, Xu et Wang [120] ont montré que la méthode peut présenter des phénoménes de fuites numériques du fluide vers le corps solide. En outre, cette méthode s'adresse principalement à la modélisation de structures minces car cela suppose que son mouvement est régi principalement par l'influence du fluide environnant. Dans le cas inverse (une structure épaisse), ce serait davantage le corps solide qui modifierait l'écoulement, et la contribution des forces fluides serait ramenée au même plan que celle des forces internes au corps. De plus, la méthode de frontières immergées originelle n'était pas non plus destinée à gérer des structures infiniment minces qui peuvent faire intervenir un saut de contraintes lors du passage du fluide au corps solide. Ce point précis fut traité, entre autres, par LeVeque (1994) [64] qui introduisit des conditions de saut et présenta ainsi la méthode d'interfaces immergées.

La principale difficulté dans les méthodes de frontières immergées se présente lorsqu'il s'agit d'imposer les conditions de bords sur la frontière. Cela apparaît relativement simple pour les maillages adaptés, mais ce n'est pas le cas pour les grilles non adaptées. C'est également cette étape qui différencie une méthode de frontières immergées d'une autre.

Considérons un domaine Ω se décomposant en un domaine fluide Ω_f et un domaine solide Ω_s dans lequel se présente un écoulement incompressible visqueux sur un corps immergé. On notera Γ l'interface fluide-structure telle que $\Gamma = \Omega_f \cap \Omega_s$. L'écoulement est modélisé par les équations suivantes :

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \dfrac{1}{\rho}\nabla p - \dfrac{\mu}{\rho}\triangle \mathbf{u} = 0 & \text{dans } \Omega_f, \quad (2.1) \\ \nabla\cdot\mathbf{u} = 0 & \text{dans } \Omega_f, \quad (2.2) \\ \mathbf{u} = \mathbf{u_s} & \text{dans } \Omega_s. \quad (2.3) \end{cases}$$

où $\begin{cases} \mathbf{u} \text{ est la vitesse,} \\ p \text{ est la pression,} \\ \rho \text{ est la masse volumique du fluide,} \\ \mu \text{ est la viscosité dynamique du fluide.} \end{cases}$

L'idée génerale pour l'implémentation des méthodes de frontières immergées consiste à discrétiser l'équation (2.1) sur un maillage cartésien non conforme à la géométrie du corps solide tout en imposant la condition de bord qu'est l'équation (2.3), en modifiant l'équation (2.1).

Pour se faire, deux approches sont considérées. La première consiste à ajouter un terme dans l'équation (2.1) qui prendra la forme d'une fonction de forcing (ou d'un terme source) et qui reproduira l'effet de la frontière dans les équations. L'équation (2.1) peut alors être réécrite sous la forme :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \frac{1}{\rho}\nabla p - \frac{\mu}{\rho}\triangle \mathbf{u} = f_b \qquad \text{dans } \Omega.$$

où f_b représente la fonction de forcing.

Cette équation est ensuite dicrétisée sur la grille cartesienne et donne lieu au système discret suivant :

$$\left[\mathcal{L}\,(\mathbf{u})\right]\{\underline{U}\} = \left\{\underline{f_b}\right\} \qquad \text{dans } \Omega,$$

avec $\begin{cases} \left[\mathcal{L}\,(\mathbf{u})\right] \text{ l'opérateur discret des équations de Navier Stokes incompressibles,} \\ \{\underline{U}\} = (\mathbf{u}, p)\,, \\ \left\{\underline{f_b}\right\} \text{ la fonction de forcing discret.} \end{cases}$

Cette approche est appelée le "continuous forcing" et des méthodes s'inscrivant dans cette catégorie seront explicitées dans la section 2.1.1.

La seconde approche consiste à garder l'équation (2.1) inchangée dans le domaine continu. Les équations sont donc discrétisées normalement *a priori*. Or, précisément, c'est ici qu'intervient la frontière immergée dans cette approche. Effectivement, La présence de la frontière immergée se traduit dans la discrétisation sur les mailles proches de l'interface, discrétisation qui est ajustée pour prendre en compte la présence de l'interface. Le système linéaire qu'il convient de résoudre est donc dans ce cas-ci :

$$\left[\mathcal{L}'\,(\mathbf{u})\right]\{\underline{U}\} = \left\{\underline{r_b}\right\} \qquad \text{dans } \Omega,$$

avec $\begin{cases} \left[\mathcal{L}'\,(\mathbf{u})\right] \text{ l'opérateur discret modifié des équations de Navier Stokes} \\ \quad \text{incompressibles,} \\ \left\{\underline{r_b}\right\} \text{ les termes connus pour l'imposition des conditions de bord.} \end{cases}$

Cette approche est appelée le "discrete forcing" et des méthodes s'inscrivant dans cette catégorie seront explicitées dans la section 2.1.2.

Par conséquent, dans l'approche du "continuous forcing", le forcing est introduit avant que les équations ne soient discrétisées. Cela permet une plus grande latitude dans le choix des schémas de discrétisation, tandis que, dans l'approche du "discrete forcing", le forcing est introduit pendant la discrétisation et donc celui-ci dépend très fortement des schémas numériques choisis par l'utilisateur. Néanmoins, bien que le "continous forcing" paraisse plus attractif dans un premier abord en raison de sa grande liberté dans les schémas numériques, l'approche du "discrete forcing" permet un contrôle direct de la précision numérique, de la stabilité et de la conservation directe des propriétés du solveur, contrairement au "continuous forcing".

2.1.1 Le "Continuous Forcing".

Nous allons ici présenter quelques méthodes de frontières immergées utilisant l'approche du "continuous forcing" : la méthode originelle de frontières immergées présentée par Peskin en 1972 et la méthode de pénalisation. Notre but n'est pas de les présenter toutes mais d'en introduire plusieurs afin de voir la variété de méthodes qui ont été engendrées.

2.1.1.1 La Méthode de Frontières Immergées introduite par Peskin [84].

La méthode originelle de frontières immergées de Peskin présentée en 1972 fut développée pour la simulation couplée des écoulements sanguins et de contractions musculaires dans un coeur en activité. Dans cette méthode, l'écoulement fluide est modélisé par les équations de Navier Stokes incompressibles qui sont résolues sur une grille cartésienne fixe. La frontière immergée est représentée par un ensemble de fibres élastiques et la position de ces fibres est suivie de manière lagrangienne par un ensemble de points dépourvus de masse qui se déplacent avec la vitesse locale du fluide.

Par conséquent, si on note $\mathbf{X_k}$ la position du $k^{\text{éme}}$ marqueur lagrangien, son évolution en temps est donnée par l'équation :

$$\frac{\partial \mathbf{X_k}}{\partial t} = \mathbf{u}\left(\mathbf{X_k}, t\right). \tag{2.4}$$

L'effet de la frontière immergée sur le fluide environnant est essentiellement capturé en transmettant les contraintes de la fibre au fluide à travers un terme de forcing local donné par :

$$\mathbf{f}(\mathbf{x}, t) = \sum_k \mathbf{F}_k\left(t\right) \delta\left(\left|\mathbf{x} - \mathbf{X}_k\right|\right),$$

avec $\begin{cases} \delta \text{ la fonction de Dirac,} \\ \mathbf{f} \text{ la force exercée sur le fluide par les fibres élastiques,} \\ \mathbf{F}_k \text{ la contrainte exercée par le } k^{\text{éme}} \text{ marqueur lagrangien.} \end{cases}$

Cette contrainte peut être déterminée de multiples manières, par exemple en ayant recours à la loi de Hooke ou encore en la considérant comme dérivant de l'énergie interne du solide.

Sur le même principe, l'équation (2.4) peut être réécrite de la manière suivante :

$$\frac{\partial \mathbf{X_k}}{\partial t} = \sum_k \mathbf{u}\left(\mathbf{x}, t\right) \delta\left(\left|\mathbf{x} - \mathbf{X}_k\right|\right).$$

En conséquence, dans le domaine continu, les équations considérées sont les suivantes :

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \left(\mathbf{u} \cdot \nabla\right) \mathbf{u} + \dfrac{1}{\rho} \nabla p - \dfrac{\mu}{\rho} \triangle \mathbf{u} = \mathbf{f} & \text{dans } \Omega_f, \\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega_f, \\ \mathbf{f}(\mathbf{x}, t) = \displaystyle\int_\Omega \mathbf{F}\left(t\right) \delta\left(\left|\mathbf{x} - \mathbf{X}\right|\right) \mathrm{d}\Omega, \\ \dfrac{\partial \mathbf{X}}{\partial t} = \displaystyle\int_\Omega \mathbf{u}\left(\mathbf{x}, t\right) \delta\left(\left|\mathbf{x} - \mathbf{X}\right|\right) \mathrm{d}\Omega. \end{cases}$$

Au niveau discret, la fonction de Dirac δ doit être lissée en raison de sa très forte irrégularité par définition. On fera alors intervenir une "approximation de la fonction de Dirac" δ_h . La forme la plus communément utilisée pour l'expression de δ_h [84] [63] introduit une nouvelle fonction φ et on définit alors δ_h par :

$$\delta_h\left(\mathbf{x}\right) = \begin{cases} \dfrac{1}{h^3} \varphi\left(\dfrac{x}{h}\right) \varphi\left(\dfrac{y}{h}\right) \varphi\left(\dfrac{z}{h}\right), & \text{si } h = \triangle x = \triangle y = \triangle z, \\ \dfrac{1}{\triangle x \triangle y \triangle z} \varphi\left(\dfrac{x}{\triangle x}\right) \varphi\left(\dfrac{y}{\triangle y}\right) \varphi\left(\dfrac{z}{\triangle z}\right) & \text{sinon,} \end{cases}$$

où h, $\triangle x$, $\triangle y$, $\triangle z$ sont les pas d'espace de discrétisation 3D, x, y, z sont les composantes cartésiennes 3D de $\mathbf{x}$ et φ est définie par :

$$\varphi\left(r\right) = \begin{cases} \dfrac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right) & \text{si } |r| \leqslant 1, \\ \dfrac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right) & \text{si } 1 \leqslant |r| \leqslant 2, \\ 0 & \text{sinon,} \end{cases}$$

où $r = \dfrac{x_i}{\triangle x_i}, i = 1$ à 3 est représentatif du nombre de cellules dans chaque direction.

D'autres définitions de δ_h ont été employées [15] [90] et celle-ci reste un facteur clé des méthodes de "continuous forcing" (voir Figure 2.1).

Les méthodes s'inscrivant dans cette catégorie ont été utilisées avec succès pour simuler une grande variété de problèmes incluant la mécanique cardiaque [83], la dynamique cochléaire [15] ou encore la dynamique des bulles [112].

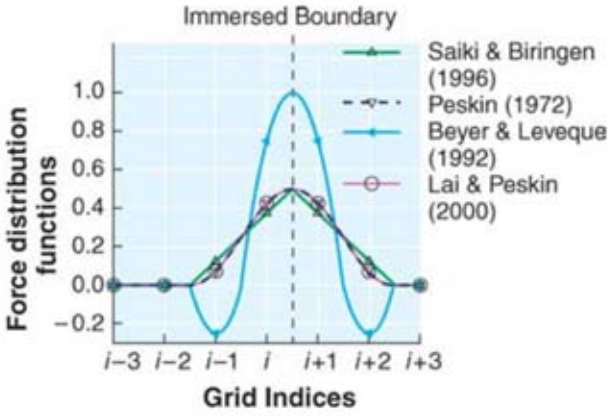


FIGURE 2.1 – Différentes définitions de δ_h [71]

Faisons enfin une remarque sur les méthodes d'interfaces immergées pour éviter une possible confusion entre les deux approches.

Bien que leur dénomination se ressemblent, les deux méthodes sont conceptuellement différentes. En effet, dans les méthodes d'interfaces immergées, les outils de marqueurs lagrangiens ou de fonction de Dirac ne sont plus indispensables ; les conditions de bord y sont imposées en modifiant les opérateurs discrets en tenant compte des conditions de saut, grâce à des interpolations faisant intervenir des développements de Taylor. Enfin, ce traitement des conditions de bord dans l'approche Interfaces Immergées rend cette méthode plus précise que l'approche présentée par Peskin.

2.1.1.2 La Méthode de Pénalisation.

L'un des inconvénients de la méthode originelle présentée par Peskin est la possibilité de fuites numériques du fluide au solide au niveau de l'interface, ce qui crée une perte de volume. Peskin et Printz [85] ont démontré que cette perte de volume était due à la non conservation de la masse au niveau des particules lagrangiennes. Autrement dit, l'équation (2.2) n'était plus vérifiée. Ils ont alors présenté une solution pour la construction d'un meilleur opérateur de divergence afin que le volume soit ainsi mieux conservé.

Une autre solution fut apportée avec la méthode de pénalisation. Cette approche fut originellement proposée par Arquis et Caltagirone [4] en 1984 qui l'appliquèrent aux équations de Navier Stokes et fut également développée par Angot *et al.* en 1989 [2] pour l'équation de Poisson.

L'approche de la méthode de pénalisation consiste à considérer l'ensemble du système (fluide+solide) comme un milieux unique et poreux dont la perméabilité varie : la perméabilité étant proportionnelle à la porosité, autrement dit, plus le milieu est perméable, plus la porosité est grande. Le modèle utilisé par Arquis et Caltagirone fut un modèle de porosité de type Brinkman qui consiste à ajouter une force de la forme :

$$\mathbf{F} = K_0 \chi_s \mathbf{u},$$

où $K_0\chi_s$ représente la perméabilité du milieu, étant donc infiniment petite dans le solide et infiniment grande dans le fluide (Cette formulation impose une vitesse nulle au

solide).

Cette force additionnelle peut également être vue comme un terme de Darcy, lui donnant un sens physique. Angot *et al.* démontrèrent en 1999 [3] que le système d'équations (2.1) à (2.3) peut être réécrit comme :

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \dfrac{1}{\rho}\nabla p - \dfrac{\mu}{\rho}\triangle\mathbf{u} = K\chi_s\,(\mathbf{u}_s - \mathbf{u}) & \text{dans } \Omega, \\ \nabla\cdot\mathbf{u} = 0 & \text{dans } \Omega, \end{cases}$$

où $\mathbf{u}_s$ est la vitesse que l'on souhaite imposer dans le solide.

En pratique $K = \dfrac{\mu}{\rho}K_0$ est un nombre tres grand (par exemple, 10^8 pour la pénalisation implicite, sinon $K \sim \dfrac{1}{\triangle t}$), et χ_s est la fonction indicatrice du solide :

$$\chi_s = \begin{cases} 1 & \text{dans le solide,} \\ 0 & \text{dans le fluide.} \end{cases} \tag{2.5}$$

Ils présentèrent ainsi les équations de Navier Stokes Incompressibles pénalisées.

Une telle pénalisation de la vitesse est qualifiée de pénalisation de type L^2. Angot *et al.* présentent également dans [3] une pénalisation de type H^1 qui consiste, non plus à pénaliser uniquement la vitesse mais également toute la partie linéaire des équations de Navier Stokes, soit le terme de la vitesse en dérivée temporelle, le terme de pression et le terme visqueux. Les équations de Navier Stokes en pénalisation H^1 s'écrivent alors sur Ω :

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t}\,(1 + K\chi_s) + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \left(\dfrac{1}{\rho}\nabla p - \dfrac{\mu}{\rho}\triangle\mathbf{u}\right)(1 + K\chi_s) \\ \qquad\qquad = K\chi_s\,(\mathbf{u}_s - \mathbf{u})\,, \\ \nabla\cdot\mathbf{u} = 0. \end{cases}$$

Ils démontrèrent également que, bien qu'en théorie la pénalisation H^1 serait plus précise que la pénalisation L^2, leurs résultats numériques ne permettent pas d'aboutir à cette même conclusion, ceci étant dû à l'approximation numérique lors de la pénalisation des termes visqueux qui crée une couche limite artificielle autour du solide.

2.1.2 Le "Discrete Forcing".

Bien que l'application des méthodes de frontières immergées à des écoulements à bas nombre de Reynolds et à nombre de Reynolds modérés ait été fructueuse, une extension à des simulations faisant intervenir des nombres de Reynolds plus élevés constitue un challenge en soit à cause de la nécessité de résoudre précisément les couches limites sur les surfaces immergées.

Dans de tels cas, la précision locale de la solution tient une importance prépondérante. Pour cette raison, d'autres approches ont été mises en place, se concentrant davantage

sur la précision locale proche de la surface immergée. Ceci peut être effectué en modifiant le stencil de calcul proche de l'interface pour imposer directement les conditions de bords sur l'interface.

Nous allons ici présenter quelques méthodes de frontières immergées utilisant l'approche du "discrete forcing" : les méthodes Ghost-Cell et Cut-Cell. Une fois de plus, notre but n'est pas de les présenter toutes mais d'en introduire plusieurs afin de voir la variété de méthodes qui ont été engendrées.

2.1.2.1 La Méthode Ghost-Cell

La méthode Ghost-Cell est issue de l'évolution des approches de Mohd-Yusof [74] et Fadlun [37]. Mohd-Yusof suggéra en 1997 l'introduction d'une force dans les équations de Navier Stokes qui imposerait la vitesse désirée de manière implicite sur la frontière immergée. Basée sur une idée de "mirroir", cette force consiste, pour des conditions de bord de type Dirichlet homogène, en l'imposition sur les premières mailles solides d'une vitesse opposée à la vitesse tangentielle des mailles fluides voisines.

Fadlun *et al.* appliquèrent cette approche dans [37] au cas 3D sur une grille MAC (marker-and-cell) avec néanmoins quelques modifications. En effet, la première différence entre l'approche de Fadlun et celle de Mohd-Yusof est le fait que, dans l'approche de Fadlun, la détermination des vitesses des noeuds fluides voisins du solide s'effectue en utilisant leurs voisins immédiats fluides et la vitesse du solide désirée. De plus, la direction d'interpolation pour Fadlun varie, pouvant être aléatoirement la direction de l'axe des abscisses ou celle des ordonnées (créant parfois des situations ambigües). Par conséquent, quand Mohd-Yusof utilisait une approche qui pouvait être qualifiée de "mirroir", ce caractère n'est pas conservé par Fadlun *et al.* qui reconstruisent plutôt la vitesse du fluide aux bords du solide à partir de la vitesse que le corps solide est supposé avoir et de son voisin fluide (voir Figure 2.2).

Ces deux types de reconstruction de la vitesse furent la base de nombreux autres schémas de reconstruction ([8],[45], [124], [27]).

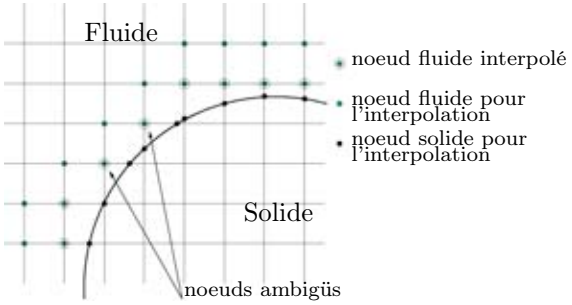


FIGURE 2.2 – Schéma d'interpolation des vitesses aux bords du solide pour Fadlun *et al.* [37]

La méthode Ghost-Cell que nous allons présenter est directement issue de Fadlun *et al.* [37] mais également de Verzicco *et al.* [114]. Il s'agit de la méthode Ghost-cell

présentée par Tseng *et al.* dans [105].

L'idée principale de leur méthode reste donc d'ajouter un terme de force dans les équations de Navier Stokes incompressibles qui imposerait les conditions de bords exactement à l'interface. Ils espèrent ainsi atteindre une représentation du système d'un ordre supérieur. Pour cela, ils introduisent une zone fantôme ("ghost cell zone") sur laquelle s'applique la force. Cette zone fantôme est constituée de l'ensemble des cellules fantômes ("ghost cells") qui sont définies telles qu'elles appartiennent au corps solide et possède au moins un voisin dans le fluide.

Afin de comprendre le principe de la méthode Ghost Cell, considérons tout d'abord un solide dont les bords coïncident avec des arêtes du maillage. On pose alors les équations de Navier Stokes incompressibles auxquelles on ajoute le terme de forcing :

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \dfrac{1}{\rho}\nabla p - \dfrac{\mu}{\rho}\triangle \mathbf{u} = \mathbf{f}\chi_{\mathtt{GZ}} & \text{dans } \Omega_f, \\ \nabla\cdot\mathbf{u} = 0 & \text{dans } \Omega_f, \end{cases}$$

où $\begin{cases} \chi_{\mathtt{GZ}} \text{ est la fonction indicatrice de la zone fantôme,} \\ \mathbf{f} \text{ est le terme de forcing.} \end{cases}$

Etant donné que l'on souhaite $\mathbf{u} = \mathbf{u}_s$ dans le solide, en considérant par exemple un schéma d'ordre 1 sur la dérivée temporelle de $\mathbf{u}$ et quelle que soit le schéma choisi en espace, on écrit :

$$\mathbf{f}^{n+1} = \frac{\mathbf{u}_s - \mathbf{u}^n}{\triangle t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \frac{1}{\rho}\nabla p - \frac{\mu}{\rho}\triangle \mathbf{u}.$$

Dès lors, sachant que l'on a choisi de prime abord un solide dont les bords coïncident avec les noeuds du maillage, on aura bien la vitesse désirée sur l'interface.

Bien entendu, pour des géométries complexes, l'interface du solide ne coïncide pas avec les noeuds du maillage. Il faut alors faire une extrapolation pour trouver la bonne vitesse à imposer aux cellules fantômes pour obtenir la vitesse souhaitée sur l'interface. L'interpolation qui en découlera sera décrite par la suite.

L'algorithme de résolution proposé par Tseng *et al.* [105] pour résoudre les équations en utilisant la méthode Ghost Cell est le suivant :

1. Détecter l'interface fluide-solide et repérer l'ensemble des cellules fantômes,
2. Effectuer l'extrapolation sur les valeurs des cellules fantômes pour imposer les conditions de bords,
3. Utiliser un schéma de type prédiction-correction pour résoudre les équations de Navier Stokes ([38] et [59] sont proposés par Tseng *et al.* , mais on peut également choisir un schéma de type Temam-Chorin [101] [28])

La première étape peut s'effectuer sans difficulté grâce à l'outil Level Set. Il s'agit donc ensuite de déterminer la bonne formule d'interpolation pour les vitesses. Pour cela on considère que l'on souhaite tout d'abord imposer des conditions de bord de type Dirichlet au solide, i.e. on veut $\mathbf{u} = \mathbf{u}_s$ dans le solide [5].

Plaçons nous pour simplifier le problème dans le cas 2D.

L'approche présentée par Tseng consiste tout d'abord à considérer le noeud fantôme ("ghost node") G et ses 2 voisins fluides les plus proches X_1 et X_2 (voir Figure 2.3). On note O le point de l'interface sur lequel on souhaite que la condition de bord soit satisfaite. Étant donné que la démarche qui suit peut être appliquée à n'importe quelle variable d'écoulement, on se place dans le cas général et on note φ la variable d'écoulement que l'on souhaite interpoler. Une interpolation linéaire 2D classique de φ est :

$$\varphi = c_0 + c_1 x + c_2 y.$$

Les coefficients inconnus c_0, c_1 et c_2 peuvent ensuite être déterminés en termes des valeurs de la variable d'écoulement sur les trois noeuds O, X_1 et X_2 (voir Figure 2.3 :

$$\{\underline{C}\} = [\mathcal{B}]^{-1} \{\underline{\varphi}\},$$

$$\text{avec} \begin{cases} \{\underline{C}\}^T = (c_0, c_1, c_2) \\ \quad \text{le vecteur contenant les trois coefficients inconnus,} \\ \{\underline{\varphi}\}^T = (\varphi_0, \varphi_1, \varphi_2) \\ \quad \text{les valeurs de la variable d'écoulement sur les trois noeuds,} \\ [\mathcal{B}] = \begin{pmatrix} 1 & x|_0 & y|_0 \\ 1 & x|_1 & y|_1 \\ 1 & x|_2 & y|_2 \end{pmatrix}. \end{cases}$$

Une fois que les coefficients sont identifiés, on peut connaître la valeur de φ sur le noeud fantôme. Faisons deux remarques. Tout d'abord, notons que dans le cas où le ghost point ne possède qu'un seul voisin fluide, alors l'interpolation utilisera uniquement ces deux points et ne dépendra plus des variables x et y mais portera uniquement sur x. De plus, dans le cas où la matrice $[\mathcal{B}]$ n'est pas inversible, cela signifie que l'un des voisins X_1 ou X_2 est situé dans la direction normale à l'interface. Dès lors, l'approche de résolution choisie est similaire à celle utilisant le point symétrique au noeud fantôme décrite ci-après.

L'inconvénient majeur de cette méthode est l'instabilité numérique rencontrée lorsque l'un des points fluides est trop proche de l'interface. En effet, les coefficients peuvent alors devenir très grands et négatifs. La valeur alors trouvée pour φ_G peut être trop grande devant celle de ses points voisins, et la solution trouvée ne converge donc pas.

Deux solutions sont proposées par Tseng pour résoudre ce problème. La première solution est de, non plus utiliser deux voisins fluides et un point de l'interface pour faire l'interpolation mais d'employer le point de l'interface O et le point symétrique du noeud fantôme G par rapport à l'interface. On notera le point symétrique I. Dès lors, on a $\varphi_G = 2\varphi_O - \varphi_I$ à l'ordre 2, où φ_I est évalué avec la méthode d'interpolation présentée précédemment (voir Figure 2.4a).

5. En pratique, l'imposition $\mathbf{u} = \mathbf{u}_s$ à l'interface est une condition suffisante.

La seconde solution est de simplement modifier localement la position de l'interface en la déplaçant afin de la faire passer par le point fluide très proche de l'interface (voir Figure 2.4b). Étant donné que, numériquement, la frontière est approximée comme localement linéaire, la précision est très peu affectée. Gibou *et al.* [44] ont montré que cette approche pouvait donner le second ordre dans la résolution de l'équation de Poisson sur un domaine irrégulier.

Une telle procédure peut également s'appliquer à des conditions de Neumann. Par exemple, supposons que l'on veut imposer des conditions de bord de type Neumann homogène pour la pression au niveau de l'interface fluide-solide, cela se traduit au niveau du point de l'interface O par :

$$\left.\frac{\partial p}{\partial \mathbf{n}}\right|_0 = 0$$

Les seules modifications qui seront apportées à la démarche exposée précédemment se situent dans les coefficients de la matrice $[\mathcal{B}]$. Définissons tout d'abord θ_0 l'angle entre l'axe des abscisses et la normale extérieure à l'interface $\mathbf{n}$[6] (voir Figure 2.3). De surcroît, on écrivait précédemment pour des conditions de Dirichlet

$$\begin{cases} \varphi\left(x|_0, y|_0\right) = c_0 + c_1 x|_0 + c_2 y|_0, & (2.6) \\ \varphi\left(x|_1, y|_1\right) = c_0 + c_1 x|_1 + c_2 y|_1, & (2.7) \\ \varphi\left(x|_2, y|_2\right) = c_0 + c_1 x|_2 + c_2 y|_2. & (2.8) \end{cases}$$

Pour des conditions de Neumann, on utilisera également les égalités (2.7) et (2.8) mais on dérivera l'égalité (2.6) pour obtenir le système suivant :

$$\begin{cases} \dfrac{\partial \varphi}{\partial \mathbf{n}}\left(x|_0, y|_0\right) = c_1 cos\left(\theta_0\right) + c_2 sin\left(\theta_0\right), \\ \varphi\left(x|_1, y|_1\right) = c_0 + c_1 x|_1 + c_2 y|_1, \\ \varphi\left(x|_2, y|_2\right) = c_0 + c_1 x|_2 + c_2 y|_2. \end{cases}$$

Ce qui s'écrit également sous la forme :

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = [\mathcal{B}]^{-1} \begin{pmatrix} \left.\frac{\partial \varphi}{\partial \mathbf{n}}\right|_0 \\ \varphi|_1 \\ \varphi|_2 \end{pmatrix}$$

$$\text{avec } [\mathcal{B}] = \begin{pmatrix} 0 & cos\left(\theta_0\right) & sin\left(\theta_0\right) \\ 1 & x|_1 & y|_1 \\ 1 & x|_2 & y|_2 \end{pmatrix}.$$

Enfin, notons que puisque nous avons défini une méthode Ghost-Cell pour imposer des conditions de bord de type Dirichlet et Neumann, il est donc tout à fait possible de faire

6. Notons que θ_0 et $\mathbf{n}$ sont des grandeurs qui peuvent être caclulées directement grâce à l'outil Level Set.

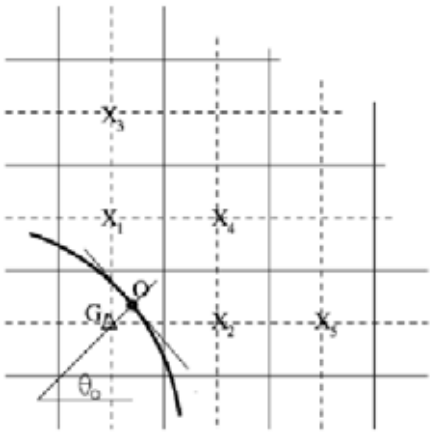


FIGURE 2.3 – Représentation d'une frontière immergée, du ghost node G et des 2 points d'interpolation X_1 et X_2 utilisés par Tseng *et al.* pour imposer des conditions de bord [105].

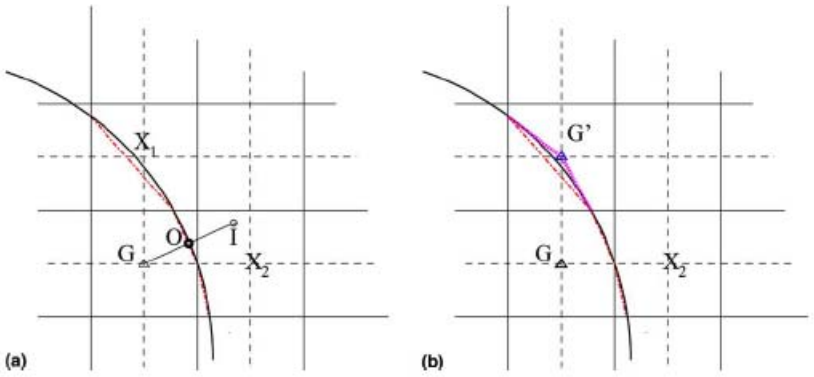


FIGURE 2.4 – Traitement particulier appliqué à la méthode de Tseng *et al.* pour éviter les instabilités numériques [105] : (a) utilisation du point image I ; (b) modification locale de la frontière immergée.

la même démarche pour des conditions mixtes de type Robin, en découplant simplement la partie de type Dirichlet et celle de type Neumann.

Les interpolations présentées jusqu'à présent sont linéaires, mais elles pourraient également être quadratiques, ce qui permettrait de garder un ordre second. De plus, une reconstruction linéaire est, certes, acceptable pour des écoulements linéaires ou pour des écoulements à haut nombre de Reynolds à condition que le premier point de grille soit situé dans la couche visqueuse [57], mais dans le cas général, à haut nombre de Reynolds, une reconstruction linéaire peut mener à des erreurs de prédiction. C'est pour de tels cas qu'une reconstruction d'ordre élevé est privilégiée.

Tseng *et al.* [105] présentent le cas bidimensionnel de l'interpolation quadratique. Mittal *et al.* [70] nous proposent l'application au cas tridimensionnel. Celle-ci s'établit alors sur huit (quatre en 2D) points d'interpolation et s'écrit :

$$\varphi(x,y,z) = c_1xyz + c_2xy + c_3yz + c_4xz + c_5x + c_6y + c_7z + c_8.$$

Les huit coefficients c_i sont ensuite déterminés de la même manière que précédemment, i.e. en utilisant les valeurs de φ aux noeuds 1 à 8 :

$$\{\underline{C}\} = [\mathcal{V}]^{-1}\{\underline{\varphi}\}, \tag{2.9}$$

où
- $\{\underline{C}\}^T = (c_1, ..., c_8)$ le vecteur contenant les huit coefficients inconnus,
- $\{\underline{\varphi}\}^T = (\varphi_1, ..., \varphi_8)$ les valeurs de la variable d'écoulement sur les huit noeuds,
- $[\mathcal{V}]$ est la matrice de Vandermonde.

$[\mathcal{V}]$ s'écrit donc :

$$[\mathcal{V}] = \begin{pmatrix} xyz|_1 & xy|_1 & xz|_1 & yz|_1 & x|_1 & y|_1 & z|_1 & 1 \\ xyz|_2 & xy|_2 & xz|_2 & yz|_2 & x|_2 & y|_2 & z|_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ xyz|_8 & xy|_8 & xz|_8 & yz|_8 & x|_8 & y|_8 & z|_8 & 1 \end{pmatrix}.$$

Mittal *et al.* utilisent également dans [70] les outils du point image et du point d'intersection à l'interface (voir Figure 2.5). Ils mettent de surcroît en évidence le fait qu'il n'y a pas toujours un unique point d'intersection à l'interface ou encore que ce point d'intersection n'est pas toujours identifiable (voir Figure 2.6). Pourtant cette identification est cruciale pour l'interpolation car celle-ci joue sur la robustesse de la solution.

Pour résoudre ces situations, Mittal *et al.* proposent l'approche suivante : ils identifient dans un premier temps le noeud de l'élément de surface qui est le plus proche de la cellule fantôme. Ensuite, ils identifient l'ensemble des éléments de surface qui partagent

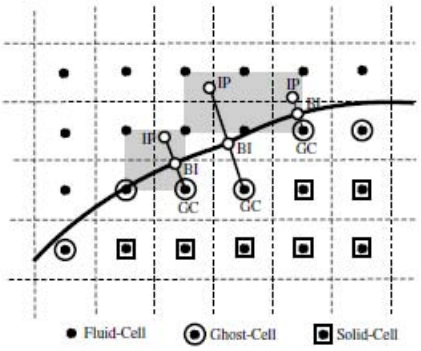


FIGURE 2.5 – Représentation d'une frontière immergée et des points utilisés par Mittal *et al.* pour leur méthode Ghost Cell [70]. GC désigne le noeud fantôme ("Ghost Cell"), BI désigne le point d'intersection à l'interface dans la direction normale ("Boundary Intersect") et IP représente le point image du noeud fantôme ("Image Point")

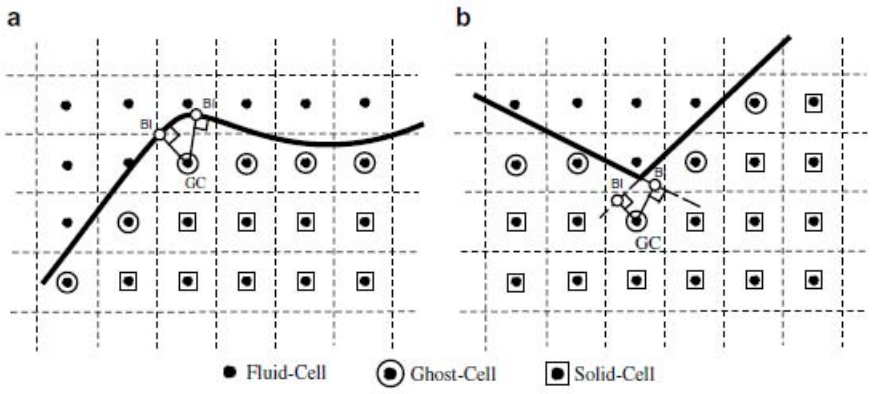


FIGURE 2.6 – Représentation des différents cas de figure pouvant poser problème pour la détermination du point d'intersection à l'interface : (a) deux points d'intersection sont trouvés, (b) le point d'intersection n'est pas sur la frontière [70]

ce noeud. Ils cherchent alors l'intersection à l'interface suivant la normale à la frontière pour chacun de ces éléments. Dans le cas où de multiples points d'intersection sont trouvés, ils choisissent le point qui a la plus petite distance à la cellule fantôme. Dans le cas où aucun point d'intersection n'est trouvé, ils répètent d'abord la recherche des éléments de surface qui partagent le noeud sur une plus grande région. Si la recherche est encore une fois infructueuse, ils inversent le premier ensemble d'éléments en choisissant comme point d'intersection le point de cet ensemble qui est le plus près de la cellule fantôme (en n'oubliant pas que ce point peut être sur les bords ou un coin d'un élément de surface).

Leur méthode, bien que non triviale, est néanmoins très robuste et leur permet de gérer des géométries complexes et en mouvement.

La méthode Ghost-Cell a été utilisée pour la simuation d'une grande variété d'écoulements, incluant les écoulement compressibles sur un cylindre ou une aile d'avion [43], des propulsions aquatiques [72] ou encore des écoulements sur des véhicules [58].

Enfin, faisons une dernière note sur la distinction entre la méthode Ghost-Cell et la méthode Ghost-Fluid [86] afin d'éviter toute confusion.

La méthode Ghost-Fluid est tout d'abord généralement utilisée pour gérer des interfaces fluide-fluide. Elle impose de manière implicite les conditions de bord à l'interface en construisant un fluide virtuel ("ghost fluid"). Ce ghost fluid est composé de ghost cells qui ont chacune une masse, une vitesse, une pression, une énergie, et sont reliées à une cellule précise du fluide réel.

Dans la méthode Ghost-Cell, chaque cellule fluide ne possède pas nécessairement une ghost cell qui lui est associée. De plus, la manière dont est évaluée la variable d'écoulement (vitesse, pression..) sur le ghost node est différente. En effet, la méthode Ghost-Fluid effectue ses interpolations suivant les axes du maillage cartésien tandis que pour la méthode Ghost Cell, il a été vu que, souvent, la direction normale à l'interface était privilégiée. Un inconvénient de la méthode Ghost Fluid est le fait que son interpolation suivant les axes cartésiens peut parfois conduire à des difficultés particulières lors de l'imposition de conditions de bord de type Neumann, chose qui, comme nous l'avons vu, peut tout à fait être gérée avec la méthode Ghost-Cell.

2.1.2.2 La Méthode Cut-Cell

Aucune des méthodes de frontières immergées présentées jusqu'à présent n'étaient façonnées pour satisfaire les lois de conservations pour les cellules situées dans le voisinage de la frontière immergée. La conservation, stricte, locale et globale de la masse et de la quantité de mouvement ne peut être garantie qu'en ayant recours à une approche Volume Finis. C'est là la première motivation pour la création de la méthode "Cut-Cell" (ou méthode de grilles cartésiennes). Cette approche fut introduite pour la première fois en 1986 par Clarke *et al.* [29] pour la modélisation d'écoulements non visqueux [111] [109] [122].

Pour la description de cette méthode, nous distinguerons deux configurations : le cas d'une grille à variables colocalisées et le cas d'une grille de type MAC.

Dans le cas d'une grille où toutes les variables sont définies aux centres des mailles, nous nous baserons sur la méthodologie de Ye *et al.* [122].

Dans un premier temps, Ye *et al.* identifient les cellules de la grille de maillage qui sont coupées par l'interface. Étant donné que la représentation de l'interface dans une maille peut être considérée comme un segment, ils identifient également les points d'intersection entre l'interface et les faces de la maille.

Pour le cas des cellules coupées par l'interface immergée et dont le centre se situe dans le fluide, ces cellules sont remodelées en retirant la partie appartenant au solide. Concernant les cellules coupées par l'interface immergées et dont le centre se situe dans le solide, ces cellules sont absorbées par leurs cellules fluides voisines. Cette redéfinition des cellules donne ainsi naissance à des cellules de forme trapézoïdale (voir Figure 2.7) (Pour davantage de détails sur le procédé de redéfinition des mailles, se référer à [107] [110]).

Le problème clé est maintenant de déterminer les flux (diffusion, convection, masse et gradient de pression) au travers des mailles trapézoïdales, tout en conservant l'ordre global du solveur. Leur solution fut l'utilisation d'une interpolation polynomiale dans une région correctement choisie et d'évaluer les flux à partir de cette fonction d'interpolation. Par exemple, considérons une variable d'écoulement φ dont on cherche à estimer le flux f sur une face quelconque. La variable φ est alors interpolée comme :

$$\varphi(x,y) = c_1x^2 + c_2y^2 + c_3xy + c_4x + c_5y + c_6$$

où, comme précédemment, $\{\underline{C}^T\} = \{c_1, ..., c_6\}$ sont les coefficients inconnus.

Cette expression de φ permet de conserver le second ordre .

De la même manière que dans la méthode Ghost-Cell, les coefficients $\{c_i\}$ sont évalués en résolvant :

$$\{\underline{C}\} = [\mathcal{V}]^{-1}\{\underline{\varphi}\},$$

avec $\{\underline{\varphi}\}^T = (\varphi_1, ..., \varphi_6)$

et $[\mathcal{V}]$ s'écrit donc :

$$[\mathcal{V}] = \begin{pmatrix} x^2|_1 & y^2|_1 & xy|_1 & x|_1 & y|_1 & 1 \\ x^2|_2 & y^2|_2 & xy|_2 & x|_2 & y|_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^2|_6 & y^2|_6 & xy|_6 & x|_6 & y|_6 & 1 \end{pmatrix},$$

où les points 1 à 6 correspondent aux six points où la valeur φ est connue : quatre points de maille et deux points de frontière (voir Figure 2.8b).

Udaykumar *et al.* [109] appliquèrent l'idée de Ye *et al.* pour des corps en mouvement. De cette manière, cette méthode a été utilisée pour simuler entre autres des objets en

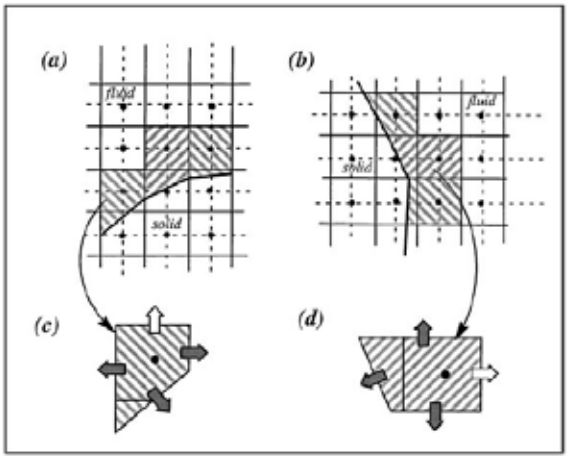


FIGURE 2.7 – Schéma représentant la redéfinition des cellules fluides au voisinage de l'interface immergée pour la méthode Cut-Cell [122]

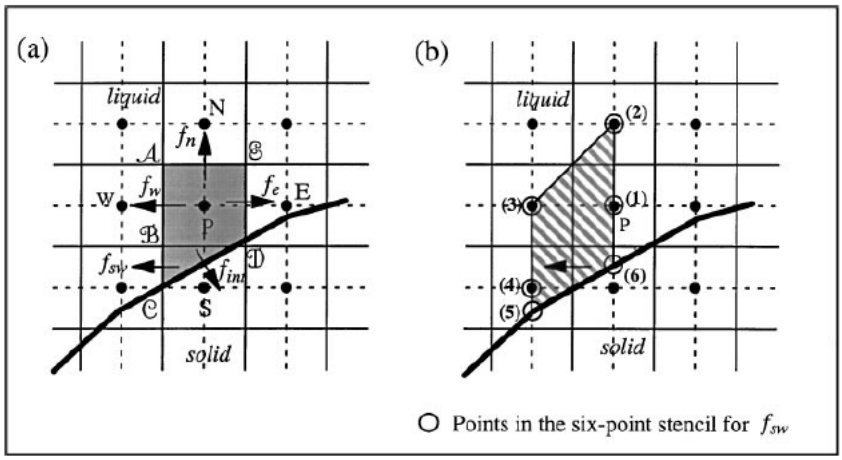


FIGURE 2.8 – Schéma représentant l'ensemble des points utilisés pour la détermination du flux $f_{\mathtt{sw}}$ pour la méthode Cut-Cell [122]

chute libre dans un fluide [72], des problèmes de solidification ou d'interactions fluides structures [108].

Ainsi, Ye *et al.* ont proposé un nouveau procédé d'interpolation pour la méthode Cut-Cell qui permet de conservé l'ordre 2 spatial du solveur et qui satisfait la conservation de la masse et de la quantité de mouvement. Ils l'appliquèrent à la simulation bidimensionnelle d'un écoulement de Stokes en présence d'un cylindre immergé, et obtinrent effectivement de l'ordre 2. Notons néanmoins que, du fait que cet écoulement ne fasse pas intervenir de termes visqueux, leur méthode n'est pas validée pour les équations de Navier Stokes. Il existe d'autres travaux sur la méthode Cut-Cell appliquée aux équations de Navier Stokes. On trouvera par exemple les travaux de Tau [100] en 1994, ou ceux de Tucker et Pan [106] en 2000. Mais toutes ces méthodes n'ont traité que des écoulements 2D. En effet, un inconvénient de la méthode Cut-Cell est la fusion des cellules car cela entraîne généralement une augmentation de la complexité lors des calculs de flux pour les cellules obliquement adjacentes. C'est pour cette raison qu'une telle procédure pose de nouvelles difficultés lors du passage en 3D.

Comme nous l'avons mentionné précédemment, il arrive que le choix d'une grille décalée (par exemple une grille MAC) soit privilégiée (voir Figure 2.9). L'utilisation d'une grille décalée pour résoudre les équations de Navier Stokes présente un certain nombre d'avantages. En effet, les grilles décalées permettent un meilleur couplage pression-vitesse. Toutefois, l'utilisation de maillage décalé présente également des difficultés, notamment lorsque l'on est en présence d'une frontière immergée sur la grille. Dans ces cas-là, considérant que les mailles associées à la pression ainsi qu'à chacune des composantes de la vitesse sont différentes, les mailles coupées par l'interface auront donc des formes différentes suivant la variable considérée. Un schéma numérique Cut-Cell doit donc gérer cette complexité supplémentaire.

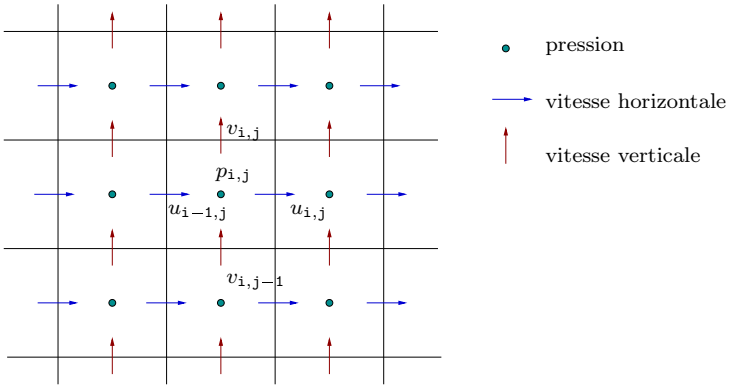


FIGURE 2.9 – Représentation d'une grille MAC

Lors de l'utilisation de grilles décalées telles que les grilles MAC, la construction des mailles et la position des variables s'effectuent en respectant une structure en cinq points (voir Figure 2.9). Lorsqu'un solide est immergé dans la grille, une telle structure en cinq points ne peut être respectée, d'autant plus si l'on souhaite utiliser la méthode de fusion des cellules vue précédemment. Ainsi, Cheny et Botella [25] [26] ont proposé en 2006 une

nouvelle méthode Cut-Cell, appelée la méthode LS-STAG, basée sur un schéma MAC, conservant la structure en cinq points ("staggered grid", d'où "STAG") et satisfaisant les lois de conservation au niveau discret. Ils utilisent la notion de distance signée au travers de l'outil Level Set (d'où "LS") pour définir l'interface fluide-solide. Cette approche fut également utilisée par Bouchon *et al.* [17] en 2012 qui parvinrent à obtenir l'ordre second globalement pour leur schéma, mais l'ordre un localement. Ils l'appliquèrent à la simulation d'un écoulement 2D sur un cylindre fixe ainsi que sur une aile d'avion.

La démarche de détermination des flux par la méthode LS-STAG utilisée par Bouchon *et al.* [17] est décrite ci-après.

Tout d'abord, considérons une grille MAC.

On considère également la présence d'un solide immergé dans un fluide. On définit la fonction level set ϕ associée :

$$\phi(i,j) = \begin{cases} -d(\mathbf{x}(i,j), \Gamma) & \text{si } \mathbf{x}(i,j) \in \Omega_s, \\ +d(\mathbf{x}(i,j), \Gamma) & \text{si } \mathbf{x}(i,j) \notin \Omega_s, \end{cases}$$

où $\begin{cases} d \text{ désigne la fonction distance} \\ \Gamma \text{ représente l'interface fluide-structure,} \\ \Omega_s \text{ représente l'ensemble du domaine solide.} \end{cases}$

L'idée principale du schéma LS-STAG est de, non plus interpoler les variables d'écoulement pour l'évaluation des flux comme on le faisait précédemment, mais de déplacer les variables de vitesse qui étaient auparavant situées aux centres des faces des mailles pour les positionner dorénavant aux centres de la partie fluide de la face. La pression n'est quant à elle pas repositionnée et reste définie aux centres des cellules de la grille (voir Figure 2.10).

En outre, on définit les ratios $r^x_{\mathtt{ij}}$ et $r^y_{\mathtt{ij}}$ qui correspondent à la proportion de face de la maille restant fluide dans chaque direction (voir Figure 2.10). Par conséquent, le ratio $r^x_{\mathtt{ij}}$ s'écrit :

$$r^x_{\mathtt{ij}} = \begin{cases} 0 & \text{si } \phi(i,j) \leqslant 0 \text{ et } \phi(i-1,j) \leqslant 0, \\ \dfrac{|\phi(i-1,j)|}{|\phi(i-1,j) - \phi(i,j)|} & \text{si } \phi(i,j) \leqslant 0 \text{ et } \phi(i-1,j) > 0, \\ \dfrac{|\phi(i,j)|}{|\phi(i-1,j) - \phi(i,j)|} & \text{si } \phi(i,j) > 0 \text{ et } \phi(i-1,j) \leqslant 0, \\ 1 & \text{partout ailleurs.} \end{cases}$$

Ensuite, ils approximent la position des points d'intersection entre l'interface immergée et les lignes de grille. Si on se place par exemple dans le cas où la cellule (i,j) est coupée par l'interface Γ en conservant son centre dans le domaine fluide (voir Figure 2.10) et que l'on note les points d'intersection de l'interface avec la cellule M_1 et M_2 tels que M_1 appartient à la droite $y = y_j$ et M_2 appartient à la droite $x = x_i$, alors M_1 et M_2 ont les coordonnées :

$$\begin{cases} M_1 : \left(x_{i-1} + r^x_{\mathtt{ij}} \triangle x \ , \ y_j\right) \\ M_2 : \left(x_i \ , \ y_{j-1} + r^y_{\mathtt{ij}} \triangle y\right) \end{cases}$$

Dès lors, on connaît la dimension des cellules coupées par l'interface et on peut repositionner les composantes de vitesse au milieu des faces coupées.

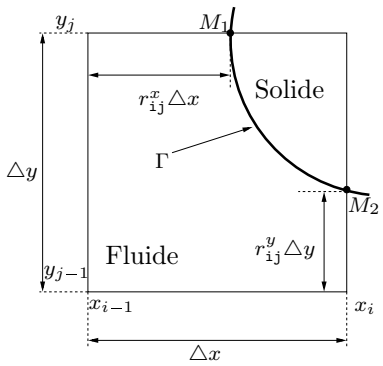


FIGURE 2.10 – Représentation d'une cellule coupée par l'interface fluide-solide

L'étape suivante consiste à discrétiser les opérateurs pour calculer les flux, en attribuant un traitement particulier pour les cellules coupées. Nous n'allons pas présenter ici la discrétisation de tous les opérateurs mais traiter le cas d'un opérateur comme exemple afin d'avoir un aperçu de la marche à suivre (pour le descriptif complet, se référer à [17] [25] [26]).

Considérons un volume de contrôle $K^{\mathtt{u}}_{\mathtt{i,j}}$ autour de la composante de vitesse $u_{i,j}$ (On définit de la même manière $K^{\mathtt{v}}_{\mathtt{i,j}}$ et $K^{\mathtt{p}}_{\mathtt{i,j}}$ les volumes de contrôle autour des variables $v_{i,j}$ et $p_{i,j}$ respectivement). On cherche à calculer $\displaystyle\int_{K^{\mathtt{u}}_{\mathtt{i,j}}} \triangle u \, \mathrm{d}\Omega$. Si $K^{\mathtt{u}}_{\mathtt{i,j}}$ est localisée loin de l'interface alors le calcul est classique et peut s'écrire au second ordre comme :

$$\frac{1}{|K^{\mathtt{u}}_{\mathtt{i,j}}|} \int_{K^{\mathtt{u}}_{\mathtt{i,j}}} \triangle u \, \mathrm{d}\Omega \approx \frac{1}{\triangle x^2} \left(u_{\mathtt{i+1,j}} - 2u_{\mathtt{i,j}} + u_{\mathtt{i-1,j}}\right) + \frac{1}{\triangle y^2} \left(u_{\mathtt{i,j+1}} - 2u_{\mathtt{i,j}} + u_{\mathtt{i,j-1}}\right)$$

Supposons maintenant que la cellule $K^{\mathtt{u}}_{\mathtt{i,j}}$ soit coupée par l'interface Γ. On note alors $K^{\mathtt{u,F}}_{\mathtt{i,j}}$ la partie fluide de $K^{\mathtt{u}}_{\mathtt{i,j}}$. Dans ce cas, on cherche à calculer $\displaystyle\int_{K^{\mathtt{u,F}}_{\mathtt{i,j}}} \triangle u d\Omega$ que l'on approxime par :

$$\frac{1}{|K^{\mathtt{u,F}}_{\mathtt{i,j}}|} \int_{K^{\mathtt{u,F}}_{\mathtt{i,j}}} \triangle u \, \mathrm{d}\Omega \approx \triangle u \left(x_i, y_{j-\frac{1}{2}}\right),$$

où $\left(x_i, y_{j-\frac{1}{2}}\right)$ correspond au centre de la cellule $K^{\mathtt{u}}_{\mathtt{i,j}}$.

Plaçons nous pour commencer dans le cas où $K^{\mathtt{p}}_{\mathtt{i,j}}$ est totalement fluide et $K^{\mathtt{u}}_{\mathtt{i,j}}$ est coupée par l'interface telle que $r^y_{\mathtt{i+1j}} \geqslant 0.5$ (voir Figure 2.11a)

Si on appelle $\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}$ la coordonnée horizontale du point d'intersection entre l'interface et la droite $y = y_{\mathtt{j}-\frac{1}{2}}$ alors on a $u_{\mathtt{i}-1,\mathtt{j}}$, $u_{\mathtt{i},\mathtt{j}}$ et $u\left(x_{\mathtt{i},\mathtt{j}-\frac{1}{2}}, y_{\mathtt{j}-\frac{1}{2}}\right)$ qui sont alignées. De plus $u\left(\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}, y_{\mathtt{j}-\frac{1}{2}}\right)$ est connue car constitue la condition de Dirichlet à la frontière fluide-solide. Par conséquent, on peut écrire l'approximation du premier ordre suivante :

$$\begin{aligned}\triangle u\left(x_i, y_{j-\frac{1}{2}}\right) \approx \frac{1}{\left(r^y_{\mathtt{i}+1\mathtt{j}}+1\right)\triangle y}\left(\frac{u_{\mathtt{i},\mathtt{j}+1}-u_{\mathtt{i},\mathtt{j}}}{r^y_{\mathtt{i}+1\mathtt{j}}\triangle y}-\frac{u_{\mathtt{i},\mathtt{j}}-u_{\mathtt{i},\mathtt{j}-1}}{\triangle y}\right)\\ +\frac{1}{\left(\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}-x_{\mathtt{i}-1}\right)}\left(\frac{u\left(\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}, y_{\mathtt{j}-\frac{1}{2}}\right)-u_{\mathtt{i},\mathtt{j}}}{\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}-x_{\mathtt{i}}}-\frac{u_{\mathtt{i},\mathtt{j}}-u_{\mathtt{i}-1,\mathtt{j}}}{\triangle x}\right)\end{aligned}$$

Cette formule n'est valable que dans le cas présenté dans la Figure 2.11a. L'utilisateur de la méthode Cut-Cell doit donc adapter les traitements à l'interface suivant la configuration dans laquelle il se trouve.

On se place maintenant dans le cas plus complexe où $K^{\mathtt{p}}_{\mathtt{i},\mathtt{j}}$ est coupée par l'interface et $r^y_{\mathtt{i}\mathtt{j}} \leqslant 0.5$ (voir Figure 2.11b).

De la même manière que précédemment, on place le point $u\left(\xi_{\mathtt{i},\mathtt{j}-\frac{1}{2}}, y_{\mathtt{j}-1}+r^y_{\mathtt{i}\mathtt{j}}\frac{\triangle y}{2}\right)$. De plus, ne pouvant plus utiliser $u_{\mathtt{i},\mathtt{j}+1}$, on introduira à la place $u\left(x_{\mathtt{i}}, y_{\mathtt{j}-1}+r^y_{\mathtt{i}\mathtt{j}}\triangle y\right)$.

Par conséquent, *a priori*, les cinq points utilisés sont :

$$\begin{cases} M_1 : \left(x_{i-1} \ , \ y_{j-\frac{1}{2}}\right) \\ M_2 : \left(x_i \ , \ y_{j-\frac{3}{2}}\right) \\ M_3 : \left(x_i \ , \ y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\frac{\triangle y}{2}\right) \\ M_4 : \left(\xi_{i,j-\frac{1}{2}} \ , \ y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\frac{\triangle y}{2}\right) \\ M_5 : \left(x_i \ , \ y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\triangle y\right) \end{cases}$$

Toutefois, étant donné que les points M_1, M_3 et M_4 ne sont pas alignés, un sixième point est nécessaire pour obtenir une interpolation du premier ordre. Le point choisi par Bouchon *et al.* [17] est le point le plus proche de l'interface, en l'occurrence $\left(x_{\mathtt{i}+1}, y_{\mathtt{j}-\frac{1}{2}}\right)$. L'interpolation pour déterminer $\triangle u\left(x_i, y_{j-\frac{1}{2}}\right)$ est donc au final :

$$\begin{aligned}\triangle u\left(x_i, y_{j-\frac{1}{2}}\right) \approx a_1 u\left(x_{i-1}, y_{j-\frac{1}{2}}\right)+a_2 u\left(x_i, y_{j-\frac{3}{2}}\right)+a_3 u\left(x_i, y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\frac{\triangle y}{2}\right)\\ +a_4 u\left(\xi_{i,j-\frac{1}{2}}, y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\frac{\triangle y}{2}\right)+a_5 u\left(x_i, y_{j-1}+r^y_{\mathtt{i}\mathtt{j}}\triangle y\right)\\ +a_6 u\left(x_{\mathtt{i}+1,\mathtt{j}-\frac{1}{2}}, y_{\mathtt{j}-\frac{1}{2}}\right).\end{aligned}$$

Les coefficients $\{a_i, i = 1..6\}$ sont déterminés en résolvant un système linéaire à travers une démarche similaire à celle présentée par Leveque et Li [64] dans leurs travaux sur les interfaces immergées.

Ainsi, en appliquant la même démarche à chaque calcul de flux proche de l'interface, Bouchon *et al.* obtinrent de l'ordre 1 en espace pour les mailles proches de l'interface et du second ordre loin de l'interface.

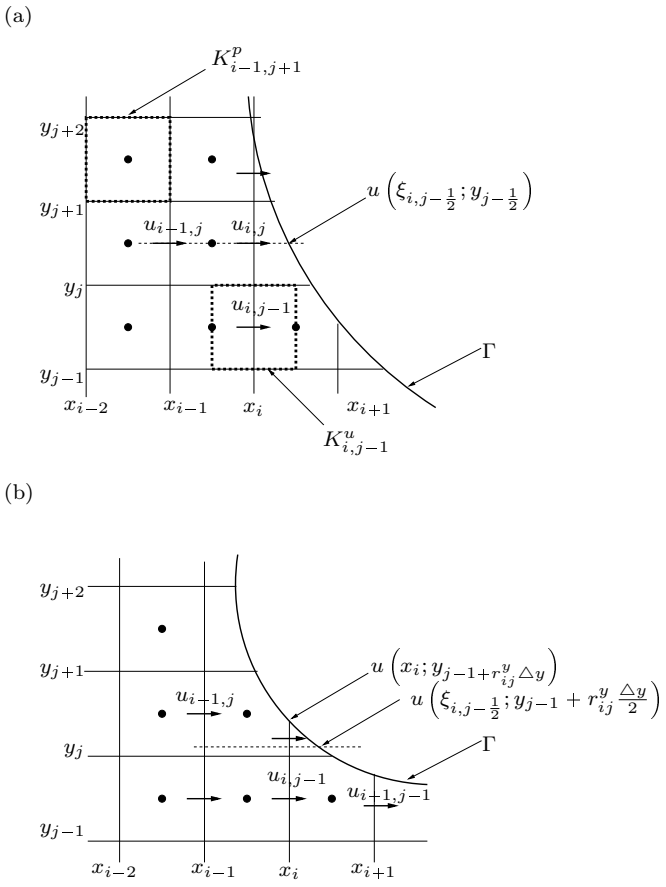


FIGURE 2.11 – Représentation d'une cellule coupée par l'interface fluide-solide

Tableau Bilan

	Ordre L^2 (global)	Ordre L^∞ (local)	Difficulté d'Implémentation	Conservation Stricte des Flux à l'interface	Difficulté pour passer au 3D
Frontières Immergées de Peskin	1	1	modérée	Non	faible
Pénalisation	1	1	faible	Non	très faible
Ghost-Cell	2	2	modérée	Non	faible
Cut-Cell colocalisées	2	2	forte	Oui	forte
Cut-Cell MAC	2	1	très forte	Oui	très forte

TABLE 2.1 – Tableau schématique comparant les différentes méthodes présentées. Les ordres sont donnés sur les vitesses uniquement et sont issus de [63] [30] [70] [122] [17]. Notons que l'ordre local de la méthode est directement lié à son habilité à gérer les simulations à nombre de Reynolds élevé à grilles égales.

2.2 La méthode ALE

Nous avons vu dans le chapitre 1 les deux descriptions classiques du mouvement dans la mécanique du continu : la description lagrangienne et la description eulérienne. Bien que chacune des deux formulations possède ses propres avantages, elles ont également leurs défauts. En effet, la méthode lagrangienne, dans laquelle chaque noeud du maillage suit une particule lagrangienne qui lui est associée, est particulièrement utilisée en mécanique des structures. La description lagrangienne permet un suivi facile des surfaces libres et interfaces entre deux matériaux. La faiblesse de cette méthode se situe dans son incapacité à suivre les grandes déformations d'un milieu sans avoir recours à des opérations de remaillage fréquentes. D'un autre côté, la description eulérienne est fréquemment utilisée en dynamique des fluides. Dans cette méthode le maillage est fixe et les grandes déformations peuvent être gérées mais généralement au prix d'une définition précise de l'interface. Ce sont ces caractéristiques, *pros and cons*, qui ont motivé le développement d'une méthode hybride combinant au mieux les meilleurs aspects des deux approches. Une telle méthode est appelée "Arbitrary Lagrangian Eulerian method", soit, méthode ALE. Cette approche fut développée originellement dans les années 1960 - 1970 entre autres par Noh [78], Franck et Lazarus [40], Trulio [103] et Hirt *et al.* [51].

Dans l'approche ALE, les noeuds du maillage peuvent être déplacés, comme dans la méthode lagrangienne, ou maintenus fixes, à la manière eulérienne ou enfin, ils peuvent être déplacés d'une manière spécifique, redéfinissant le maillage (voir Figure 2.12). Grâce à cette liberté dans le déplacement du maillage dans la méthode ALE, de plus grandes déformations peuvent être gérées par rapport à la description lagrangienne, avec une plus grande résolution que ne l'offrirait la description purement eulérienne. La figure 2.13 illustre la capacité de l'approche ALE à gérer des déformations significatives du maillage de calcul, tout en préservant une description lagrangienne de l'interface. Dans cet exemple [35], un maillage d'éléments finis est utilisé pour modéliser la déformation par une charge explosive d'un tuyau cylindrique rempli d'eau.

Notre but ici est de présenter une étude de la méthode ALE, d'introduire ses aspects conceptuels et son application à la dynamique des fluides. Nous poserons donc les bases de la description ALE, puis nous présenterons les équations fondamentales reformulées en ALE et la redéfinition du maillage qui est nécessaire dans cette approche. Enfin, nous l'appliquerons au cas de l'interaction fluide-structure. Pour se faire, nous suivons la présentation faite de la méthode dans [35].

2.2.1 L'état de référence

Considérons tout d'abord deux configurations : une configuration matérielle (ou lagrangienne) $\mathcal{R}_X$, et une configuration spatiale (ou eulérienne) $\mathcal{R}_x$. Étant donné que la méthode ALE empreinte des caractéristiques aux deux formulations, aucune des deux configurations n'est prise comme référence. On introduit donc une troisième configuration, la configuration de référence $\mathcal{R}_\chi$ où les coordonnées de référence χ sont introduites pour suivre les points de grille. Il ne faudra donc pas confondre la grille spatiale, fixe qui correspond au maillage eulérien, l'ensemble des particules lagrangiennes qui constitue le maillage lagrangien et enfin la grille de référence. Afin de pouvoir passer d'une configu-

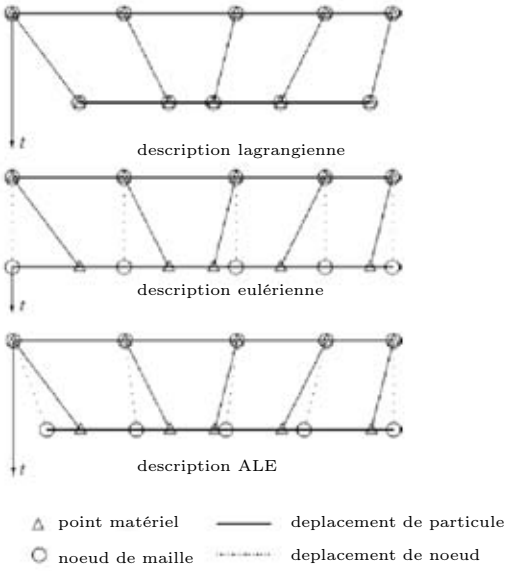


FIGURE 2.12 – Grille associée à chaque description [35]

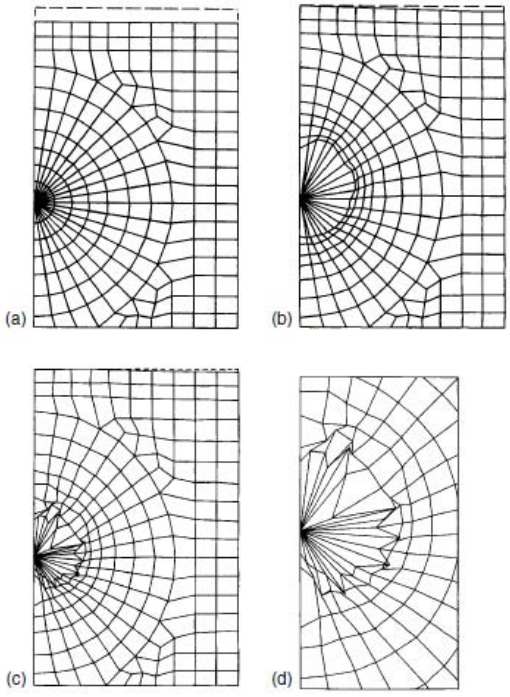


FIGURE 2.13 – Comparaison de l'évolution des maillages lagrangien et ALE : (a) maillage éléments finis initial ; (b) maillage ALE à $t = 0.1$ms ; (c) maillage lagrangien à $t = 0.1$ms ; (d) zoom sur le maillage lagrangien [35]

ration à une autre, on introduit trois applications ϕ, φ et ψ. Les domaines matériel et spatial sont liés par l'application ϕ ; le domaine de référence est lié aux domaines matériel et spatial par les applications ψ et φ respectivement (voir Figure 2.14). Elles sont donc définies par :

$$\begin{aligned} \phi(.,t): \quad & \mathcal{R}_X \to \mathcal{R}_x \\ & \mathbf{X} \mapsto \mathbf{x} = \phi(\mathbf{X},t), \end{aligned}$$

$$\begin{aligned} \varphi(.,t): \quad & \mathcal{R}_\chi \to \mathcal{R}_x \\ & \chi \mapsto \mathbf{x} = \varphi(\chi,t), \end{aligned}$$

et,

$$\begin{aligned} \psi(.,t): \quad & \mathcal{R}_\chi \to \mathcal{R}_X \\ & \chi \mapsto \mathbf{X} = \psi(\chi,t). \end{aligned}$$

Les trois fonctions sont liées par la relation

$$\phi = \varphi \circ \psi^{-1},$$

et par conséquent ne sont pas indépendantes les unes des autres. Grâce à cette dernière relation, on s'aperçoit que l'on travaille naturellement avec l'application ψ^{-1}, nous nous concentrerons sur celle-ci plutôt que sur son inverse. ψ^{-1} est donc définie par :

$$\begin{aligned} \psi^{-1}(.,t): \quad & \mathcal{R}_X \to \mathcal{R}_\chi \\ & \mathbf{X} \mapsto \chi = \psi^{-1}(\mathbf{X},t). \end{aligned}$$

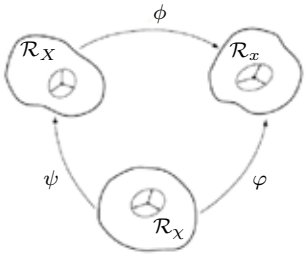


FIGURE 2.14 – Représentation des liens entre les différentes configurations [35]

Définissons de plus leurs gradients :

$$\nabla\phi = \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{X}} & \mathbf{u} \\ 0 & 1 \end{pmatrix},$$

$$\nabla\varphi = \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial \chi} & \widehat{\mathbf{u}} \\ 0 & 1 \end{pmatrix},$$

et

$$\nabla\psi^{-1} = \begin{pmatrix} \frac{\partial \chi}{\partial \mathbf{X}} & \mathbf{w} \\ 0 & 1 \end{pmatrix},$$

$$\text{où} \begin{cases} \mathbf{u} = \left.\dfrac{\partial \mathbf{x}}{\partial t}\right|_{\mathbf{X}}, \\ \widehat{\mathbf{u}} = \left.\dfrac{\partial \mathbf{x}}{\partial t}\right|_{\chi}, \\ \mathbf{w} = \left.\dfrac{\partial \chi}{\partial t}\right|_{\mathbf{X}}. \end{cases}$$

$\mathbf{u}(\mathbf{x}, t)$ correspond à la vitesse d'une particule coïncidant avec le noeud situé en $\mathbf{x}$ à l'instant t, $\widehat{\mathbf{u}}(\mathbf{x}, t)$ représente la vitesse du maillage et $\mathbf{w}$ est la vitesse des particules dans le domaine de référence. Par la suite, nous les appellerons respectivement vitesse des particules, vitesse de maillage et vitesse de référence.

Établissons maintenant le lien entre les différentes vitesses $\mathbf{u}$, $\widehat{\mathbf{u}}$ et $\mathbf{w}$. Pour se faire, on utilise la relation $\phi = \varphi \circ \psi^{-1}$ que l'on différencie :

$$\begin{aligned} \nabla\phi(\mathbf{X}, t) &= \nabla\varphi\left(\psi^{-1}(\mathbf{X}, t)\right) \nabla\psi^{-1}(\mathbf{X}, t) \\ &= \nabla\varphi(\chi, t)\, \nabla\psi^{-1}(\mathbf{X}, t). \end{aligned}$$

Grâce au passage à l'écriture matricielle, on voit apparaître la relation :

$$\mathbf{u} = \widehat{\mathbf{u}} + \frac{\partial \mathbf{x}}{\partial \chi}\mathbf{w},$$

que l'on réécrit sous la forme

$$\mathbf{c} = \mathbf{u} - \widehat{\mathbf{u}} = \frac{\partial \mathbf{x}}{\partial \chi}\mathbf{w},$$

définissant ainsi la vitesse convective $\mathbf{c}$ qui constitue la vitesse relative entre le domaine considéré et le maillage. La vitesse convective $\mathbf{c}$ ne doit pas être confondue avec la vitesse $\mathbf{w}$. En effet, comme il l'a été mentionné plus tôt, $\mathbf{w}$ est la vitesse des particules dans le domaine de référence tandis que $\mathbf{c}$ est la vitesse des particules lagrangiennes par rapport au le domaine spatial, i.e. eulérien.

Notons enfin que les deux descriptions lagrangienne et eulérienne peuvent être retrouvées en tant que cas particuliers de la description ALE. Effectivement, en choisissant $\psi = I$ où I représente l'application identité, alors $\mathbf{X} = \chi$, et la description ALE devient

la description lagrangienne. On a alors les vitesses matérielle $\mathbf{u}$ et de maillage $\hat{\mathbf{u}}$ qui coïncident, soit $\mathbf{c} = 0$. D'un autre côté, si on prend $\varphi = I$, alors $\mathbf{x} = \chi$, nous permettant ainsi de retrouver la description eulérienne. La vitesse de maillage est donc nulle et la vitesse convective $\mathbf{c}$ est égale à la vitesse matérielle $\mathbf{u}$.

2.2.2 Formulation ALE des équations de Navier Stokes Incompressibles

Pour passer d'une formulation eulérienne à une formulation ALE sur les équations de Navier Stokes incompressibles, les choses sont extrêmement simples : pour obtenir la forme ALE des équations de conservation de masse et de la quantité de mouvement, il suffit de remplacer les différents termes de convection utilisant la vitesse matérielle $\mathbf{u}$ par la vitesse convective $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$. Par conséquent, en incompressible, l'équation de conservation de la masse $\nabla \cdot \mathbf{u} = 0$ est inchangée (ce qui ne serait pas le cas en compressible), et l'équation de conservation de la quantité de mouvement étant auparavant :

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{x}} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\triangle \mathbf{u} + \mathbf{f}$$

devient

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\chi} + (\mathbf{c} \cdot \nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\triangle \mathbf{u} + \mathbf{f}.$$

Notons que les membres de droite de cette dernière équation sont écrits en formulation eulérienne et la forme ALE est uniquement présente dans le membre de gauche. La similarité entre les équations issues des deux formulations a conduit certains auteurs a appelé cette méthode la description *quasi-eulérienne* [9].

De nos jours, la majorité des codes ALE est basée sur une discrétisation spatiale soit éléments finis, soit volumes finis. L'approche volumes finis est très répandue dans le domaine de la mécanique des fluides tandis que les éléments finis sont généralement préférés pour la mécanique des solides.

Comme nous l'avons déjà mentionné plus tôt, le principal avantage de la formulation ALE est qu'elle représente une combinaison souple des descriptions eulérienne et lagrangienne. Cependant, l'implémentation de la méthode ALE requiert la formulation d'une procédure de mise à jour du maillage qui assignerait les vitesses aux noeuds du maillage à chaque instant. Cette méthode de mise à jour devrait, *a priori*, être choisie par l'utilisateur. Toutefois, l'algorithme de remaillage influence fortement la performance de la méthode ALE et peut représenter une grande contrainte pour l'utilisateur s'il n'est pas effectué automatiquement.

Il existe deux approches pour mettre à jour le maillage. La première est basée sur un concept géométrique d'une régularisation de maillage qui permet de garder le maillage aussi régulier que possible et ainsi éviter tout enchevêtrement ou chevauchement de mailles pendant le calcul. La seconde approche consiste à considérer la méthode ALE comme une technique d'adaptation de maillage pour, par exemple, concentrer davantage d'éléments

dans les zones de fort gradient, mais dès lors, une estimation correcte de l'erreur est requise comme point de départ de l'algorithme de remaillage.

2.2.3 La régularisation de maillage

L'objectif de la régularisation de maillage est de maintenir le maillage aussi régulier que possible pendant tout le temps des calculs. Cela permettra d'éviter ainsi les trop grandes distorsions du maillage et les resserrements, diminuant donc les erreurs numériques dues au maillage géométrique.

La régularisation de maillage demande que les coordonnées des noeuds mises à jour soient spécifiées à chaque itération, grâce à la vitesse de maillage $\hat{\mathbf{u}}$ par exemple. Ces procédures de mises à jour de maillage sont classées en deux catégories : suivant si on impose le déplacement à la frontière *a priori* ou si le déplacement est inconnu.

Lorsque le déplacement des surfaces matérielles est connu *a priori*, alors il en est de même pour le déplacement du maillage. Ce déplacement du maillage s'effectue en définissant une vitesse de maillage $\hat{\mathbf{u}}$ adéquate dans le domaine. Comme ce point dépend entièrement de l'utilisateur, il s'agit là d'un point délicat qui jouera sur l'efficacité de la méthode ALE. En général ce choix implique une description lagrangienne aux frontières en mouvement (le déplacement du maillage coïncidant donc avec le déplacement de la frontière) tandis que la formulation eulérienne (correspondant au choix $\hat{\mathbf{u}} = 0$) est utilisée loin de la frontière en mouvement. On définit donc une zone de transition au milieu.

Cette approche a été utilisée entre autre pour des problèmes d'interactions fluide-structure par Huerta et Liu [54], ainsi qu'en mécanique des solides non linéaires par exemple [88] [113].

Si on suppose maintenant que le déplacement de la frontière est inconnu, alors on suppose qu'au moins une partie de la surface est suivie de manière lagrangienne au cours du temps. D'où une séparation des degrés de libertés du système en degrés de libertés purement lagrangiens ($\hat{\mathbf{u}} = \mathbf{u}$) et ceux purement eulériens ($\hat{\mathbf{u}} = 0$). Cette méthode fut utilisée également pour les problèmes d'interactions fluide-strcture par Liu et Chang [65] et en mécanique des solides par Haber [47] [48]. Huerta et Liu [54] ont amélioré cette méthode en évitant la résolution des équations pour la vitesse de maillage à l'intérieur du domaine matériel et assurant le suivi des surfaces matérielles par la résolution de l'équation $\mathbf{w}.\mathbf{n} = 0$ où $\mathbf{n}$ est la normale unitaire le long des surfaces matérielles. Une fois que le déplacement des frontières est déterminé, alors les vitesses peuvent être calculées, par des interpolations par exemple, pour redéfinir le maillage.

Trois approches sont présentées dans [35] pour la régularisation de maillage : le transfinite mapping, le laplacian smoothing et le lissage de maillage. Nous ne nous étendrons pas à les décrire ici mais conseillons au lecteur intéressé de se référer à cet ouvrage.

2.2.4 L'adaptation de maillage

Lorsque l'on choisit d'utiliser la description ALE comme une méthode d'adaptation de maillage, l'objectif est alors d'optimiser le maillage pour décrire correctement les zones de fort gradient et de faible gradient, avec le plus faible coût possible. Il s'agit donc de raffiner le maillage dans les zones appropriées, en déplaçant les noeuds vers les zones de fort gradient (gardant donc le même nombre total de noeuds). L'algorithme inclut ensuite un indicateur d'erreur, et le maillage est ensuite modifié pour obtenir une equi-distribution de l'erreur dans tout le domaine. La redistribution peut se faire en utilisant des équations différentielles elliptiques ou paraboliques. Cette technique peut de surcroît être couplée avec des procédures traditionnelles de raffinement de maillage, comme la r-adaptation [7] par exemple, pour améliorer plus encore la précision [5].

Des études sur l'utilisation de la méthode ALE comme technique d'adaptation de maillage ont été effectuées dans le domaine de la mécanique des solides [55] [5] [6] [88].

L'adaptation de maillage a trouvé une grande utilisation en dynamique des fluides. Souvent une attention doit être portée sur le caractère directionnel de l'écoulement, donnant donc naissance à des procédures d'adaptation anisotropiques [39] [22] [76].

2.2.5 Méthode ALE appliquée aux interactions fluide-structure

En raison de sa grande capacité à suivre les interfaces, la méthode ALE est utilisée pour modéliser des grandes déformations ou des interactions fluide-structure. Elle est généralement appliquée à des problèmes de modélisation d'impacts, de propagation de fissures ou encore de détonation.

Concentrons nous sur le cas des interactions fluide-structure.

Le long d'une interface fluide-solide, la vitesse des particules fluides est couplée avec une structure, déformable ou rigide. L'imposition de conditions de bord cinématiques consiste à imposer qu'aucune particule ne peut traverser l'interface. Cela peut s'effectuer de manière directe en formulation ALE en utilisant son caractère lagrangien le long de l'interface. Cela revient donc à choisir $\mathbf{w} = 0$ ou $\hat{\mathbf{u}} = \mathbf{u}$. Toutefois, cette condition peut être assouplie en imposant la condition $\mathbf{w} = 0$ le long de la normale unitaire extérieure au domaine fluide, soit

$$\mathbf{w}.\mathbf{n} = 0 \quad \text{ou} \quad \hat{\mathbf{u}}.\mathbf{n} = \mathbf{u}.\mathbf{n},$$

où $\mathbf{n}$ représente la normale unitaire extérieure au domaine fluide.

Cependant, en raison du couplage entre le fluide et la structure, des conditions supplémentaires sont nécessaires pour assurer que les domaines fluide et solide ne vont pas se détacher ou se chevaucher au cours d'un déplacement. Ce couplage consiste à imposer une continuité des vitesses ou des déplacements à travers l'interface.

En pratique, deux noeuds sont placés de part et d'autre de l'interface, un noeud fluide et un noeud solide. Étant donné que le fluide est traité en formulation ALE, le mouvement

7. Il existe trois types de raffinement de maillage : la r-adaptation qui relocalise les noeuds, la h-adaptation qui ajoute des noeuds localement et la p-adaptation qui change l'ordre des schémas numériques mis en jeu.

du maillage fluide peut être choisi totalement indépendamment du mouvement du fluide lui-même. En particulier, on peut contraindre les noeuds fluides à rester contigüs aux noeuds de la structure et garder cet alignement au cours des itérations. L'alignement permanent des noeuds aux interfaces permet au couplage fluide-strucutre d'être effectué de la manière la plus simple.

Il existe une grande littérature sur les problèmes d'interactions fluide-structure en formulation ALE, aussi bien pour les structures rigides que déformables [66] [79] [91] [125] [23] [75].

2.3 Les méthodes de maillages adaptatifs

Les méthodes de maillages adaptatifs sont constituées des différentes méthodes où le maillage se raffine localement pour devenir plus précis en se subdivisant. Ces méthodes sont appelées méthodes AMR (*adaptative mesh refinement*). Le raffinement peut s'effectuer en espace et/ou en temps.

Elles sont généralement utilisées lorsque la solution présente des gradients d'amplitudes très variées ou encore lorsque les phénomènes mis en jeu se déroulent sur les échelles très différentes, en temps ou en espace.

L'approche AMR possède ainsi les avantages des approches numériques sur grilles fixes, une bonne précision et un coût de calcul faible en comparaison à un maillage fixe uniforme de précision égale.

L'inconvénient majeur de cette approche concerne la gestion de la hiérarchie entre les différentes cellules, qui augmente en complexité lorsque le nombre de niveaux de remaillage augmente (en particulier dans le cas de maillages non structurés).

Il existe trois types distincts de raffinement :
- le h-raffinement où des points de grille sont ajoutés localement,
- le r-raffinement où les points de grille sont repositionnés,
- le p-raffinement où l'ordre des schémas numériques est modifié.

Nous nous concentrerons sur le h-raffinement car celui-ci est la technique AMR la plus utilisée.

"h-raffinement" signifie raffinement hiérarchique. L'idée d'un raffinement hiérarchique a été introduite par Marsha Berger [10]. Cette approche se base sur le principe d'arbres et utilise la relation parent/enfant pour décrire le passage d'une grille de niveau n à une grille de niveau $n+1$.

On considère que la grille principale (la moins raffinée) constitue le niveau zéro et représente le tronc de la structure d'arbre. La grille la plus raffinée représente les feuilles et tous les autres niveaux de raffinement intermédiaires forment les branches de l'arbre.

Deux approches sont fréquemment utilisées :
- l'AMR de Berger,
- les quad-tree en 2D ou octree en 3D.

Les deux méthodes se différencient dans leur manière de créer les grilles raffinées à partir de la grille principale.

2.3.1 L'AMR de Berger

L'approche de Berger consiste à immerger une grille raffinée dans la grille grossière de niveau inférieure. L'utilisateur définit des critères qui, grâce à une procédure d'estimation d'erreur, permettront de déterminer si un niveau de raffinement additionnel est requis ou non, et où celui-ci doit être effectué le cas échéant. Cette formulation AMR se déroule en une résolution "niveau par niveau", qui conduit à la définition de sous itérations en temps, proportionnelles à la tailles des cellules de chaque niveau, nécessaires pour respecter la condition CFL (voir Figure 2.15).

2.3.2 Les Octrees

L'approche Octree (quad-tree en 2D) décrit une technique AMR utilisant une structure hiérarchique dans laquelle chaque élément possède exactement huit (quatre en 2D) enfants. Cette technique fut introduite pour la première fois par Dezeeuw et Powel [33] qui l'appliquèrent à des géométries complexes tels que des profils d'ailes. L'utilisation des octrees se déroulent en définissant initialement une première grille uniforme, qui sera par la suite raffinée : les cellules proches du corps ou faisant intervenir des phénomènes à échelle plus fine (par exemple, le sillage d'une aile ou les projections d'un jet) sont subdivisées afin d'atteindre une plus grande précision (voir Figure 2.16).

Enfin, terminons cette description de l'approche AMR par la présentation d'un algorithme pour le calcul d'une solution sur un niveau donné :

1. en se plaçant au niveau considéré, on supprime le maillage de l'itération précédente en l'écrasant par la création d'un nouveau bloc de maillage.
2. Ces nouveaux blocs de maillage sont initialisés par une prolongation de la solution issue du niveau parent.
3. Les conditions limites sont mises à jour.
4. La solution numérique est calculée pour l'itération en temps considérée.
5. La solution numérique sur l'écoulement parent est corrigée dans le but de préserver les propriétés de conservation du schéma numérique ([36], §4,2,4).

Dans le présent manuscrit, nous ne décrivons pas les détails de gestion des frontières entre chaque niveau de grille, ni les critères possibles de raffinement, ni le procédé de prolongation d'une solution d'un niveau de raffinement n au niveau $n+1$. Pour le lecteur intéressé, nous le renvoyons à la lecture du chapitre 4 de la thèse de Yohann Eude [36] où il trouvera la description de ces différents points.

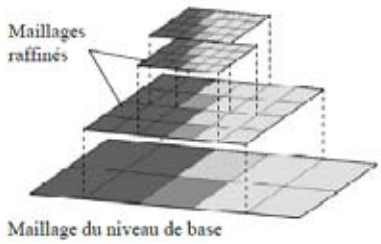


FIGURE 2.15 – Représentation des différents niveaux de hiérarchie dans le cas d'un AMR de Berger. L'échelle de gris représente la solution dans le cas d'une discontinuité [36].

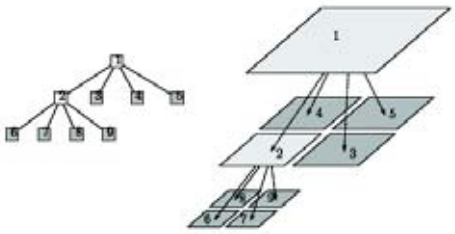


FIGURE 2.16 – Représentation d'une cellule quad-tree. Les cellules blanches sont des cellules parents et les cellules grises sont des cellules enfants [126].

2.4 Conclusion

Tout au long de ce chapitre, nous avons vu trois approches distinctes de modélisation d'interaction fluide-structure : les méthodes de type frontières immergées, la méthode ALE et les méthodes de maillages adaptatifs (ou méthodes AMR).

Les méthodes de frontières immergées se positionnent toutes sur une grille fixe indépendante du solide et se différencient ensuite en deux classes : les méthodes du "direct forcing" et du "discrete forcing". Le "direct forcing" place la présence de la frontière immergée dans les équations continues et laisse l'utilisateur libre du choix des schémas numériques. Le "discrete forcing", quant à lui, prend en considération la présence de l'interface dans la discrétisation des équations et impose donc des restrictions sur les schémas. Bien que moins souples que les méthodes du "direct forcing", nous avons pu constater à travers 2.1 que les méthodes du "discrete forcing" sont plus précises.

La méthode ALE se place quant à elle dans le cadre d'une grille adaptée à la présence du solide. Cette approche combine d'une manière souple les descriptions eulérienne et lagrangienne et bénéficie en conséquence des avantages de chacune. Elle possède donc une grande précision mais souffre de procédures de remaillage dès que le corps solide se déplace et est difficilement envisageable pour des configurations 3D.

Enfin, les méthodes AMR se positionnent initialement sur une grille principale fixe, non adaptée au solide ou à l'écoulement. Néanmoins, la grille possède la faculté de se raffiner localement, de manière dynamique si nécessaire, pour prendre en compte de manière plus précise les phénomènes de plus petites échelles. Étant sur une grille fixe, les méthodes

AMR sont transposables au 3D. Toutefois, la gestion des différents niveaux de hiérarchie amènent des difficultés non négligeables au cours de l'implémentation.

Par conséquent, chaque méthode possédant ses propres avantages et inconvénients, le choix d'une méthode plutôt qu'une autre est fonction d'un grand nombre de critères.

Tout d'abord, si l'on souhaite modéliser des phénomènes tridimensionnels, la méthode ALE ne sera généralement pas privilégiée car cette approche est très lourde à transposer au 3D, on aura alors tendance à privilégier les méthodes sur grilles fixes. Toutefois, on ne choisira pas non plus n'importe quelle méthode de frontières immergées. En effet, les méthodes Cut-Cell relèvent du défi lorsqu'il s'agit de les porter au 3D.

D'un autre côté, si l'on se place dans des cas de simulations bidimensionnelles d'objets fixes ou à faibles déplacements, la méthode ALE peut s'avérer un choix judicieux. En effet, en raison des très faibles déplacements de la structure, le maillage sera très peu modifié au cours des itérations, ne présentant donc pas un fort coût de calcul. Ainsi, l'utilisateur pourra profiter pleinement de la grande précision de la méthode ALE tout en ayant un faible coût opérationnel.

Enfin, lorsque l'on cherche à modéliser des structures 2D soumises à un écoulement et en déplacement, il revient à l'utilisateur de définir des priorités dans ses objectifs (précision, coût de calcul...) et les contraintes qu'il est prêt à s'imposer dans la programmation, afin de choisir la méthode numérique qui convient au mieux.

Dans notre cas, nous souhaitons simuler des structures 2D et 3D, mobiles, déformables, pour un coût de calcul le plus faible possible et une précision d'ordre 2 au moins global. Face à tous ces objectifs, nous avons choisi une méthode de frontières immergées sur grille cartésienne, qui nous permettra d'utiliser des schémas numériques simples et une parallélisation aisée.

Séduits par la facilité d'implémentation de la méthode de pénalisation, nous avons choisi de la combiner avec la méthode Ghost-Cell afin d'obtenir une méthode numérique d'implémentation simple et de précision supérieure à la pénalisation classique. Cette nouvelle approche sera présentée dans le chapitre 3 de ce manuscrit. Enfin, convaincus de l'efficacité de l'outil Level Set, nous l'avons choisi pour le suivi de l'interface fluide-structure.

Chapitre 3

La Méthode IPC (Image Point Correction)

Sommaire

3.1 Résolution des équations de Navier Stokes incompressibles . . . 54

3.1.1 Les équations modélisant l'écoulement fluide . . . 54

3.1.2 Discrétisation des équations . . . 54

3.2 La méthode IPC . . . 57

3.2.1 Principe de la méthode . . . 58

3.2.2 Application aux équations de Navier Stokes . . . 61

3.2.3 Cas d'un solide en mouvement . . . 63

3.3 Résultats Numériques . . . 65

3.3.1 Étude de la convergence . . . 65

3.3.2 Cas d'un cylindre, fixe ou mobile . . . 66

3.3.3 Cas de deux cylindres fixes . . . 74

3.3.4 Sédimentation d'une bille . . . 78

3.4 Conclusion . . . 80

Dans ce chapitre, nous présentons une méthode de type frontières immergées : la méthode IPC (Image Point Correction). Cette méthode est basée sur l'approche Ghost-Cell et sur la méthode de pénalisation, pour simuler des écoulements incompressibles. Nous travaillons donc sur maillage cartésien. La résolution des équations de Navier Stokes suit un schéma de type prédicteur-correcteur. Cette nouvelle approche est facilement applicable à la modélisation d'écoulements sur des corps solides, en deux ou trois dimensions, pour des structures fixes ou mobiles, rigides ou déformables.

Ainsi, après avoir présenté la méthode de résolution des équations de Navier Stokes incompressibles(§3.1), nous introduirons le principe de la méthode IPC et nous l'incorporerons dans les équations de Navier Stokes (§3.2). Enfin, après avoir établi l'ordre de cette méthode, nous la validerons via la comparaison avec des cas tests issus de la littérature, pour le cas d'un ou plusieurs corps rigide(s) fixe(s) ou en mouvement dont on impose le déplacement (§3.3.1 à §3.3.3), et par le cas de la sédimentation d'une bille (§3.3.4).

3.1 Résolution des équations de Navier Stokes incompressibles

3.1.1 Les équations modélisant l'écoulement fluide

Nous supposons être en présence d'un écoulement fluide incompressible et visqueux. Par conséquent, les équations considérées sont les équations de Navier Stokes incompressibles :

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = -\nabla p + \mu\triangle\mathbf{u}, \tag{3.1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{3.2}$$

avec ρ la masse volumique du fluide, $\mathbf{u}$ la vitesse, p la pression et μ la viscosité dynamique. On associera à ces équations des conditions de bord et conditions initiales.

Dans le but de travailler avec des équations adimensionnées, nous introduisons trois grandeurs caractéristiques : une vitesse de référence U_∞, la masse volumique de référence ρ_0 et la longueur du corps immergé D. Puis, le temps est adimensionné par l'introduction du temps de référence $t_0 = \frac{D}{U_\infty}$, de même pour la pression avec $p_0 = \rho_0 U_\infty^2$. Ainsi, les équations deviennent :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} = -\nabla p + \frac{1}{Re}\triangle\mathbf{u}, \tag{3.3}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{3.4}$$

où $Re = \frac{\rho_0 U_\infty D}{\mu}$ représente le nombre de Reynolds.

Notons que suivant les applications que nous choisirons de simuler, il pourra s'avérer plus pratique de conserver la forme dimensionnelle des équations (par exemple, pour la simulation de la chute d'une bille sous l'effet de la gravité).

3.1.2 Discrétisation des équations

Étant donné que nous travaillons sur une grille cartésienne, les équations de Navier Stokes sont discrétisées en utilisant des schémas différences finies pour les variables collocalisées $(\mathbf{u}, p)$. Nous introduirons également la vitesse $\mathbf{U}$ centrée sur les faces des mailles (correspondant à la vitesse d'un schéma MAC), suivant la définition donnée par Mittal *et al.* dans [70], dans le but de supprimer les oscillations de pression dues aux variables collocalisées et d'avoir des schémas numériques plus compactes, ce qui sera décrit ultérieurement.

On définit tout d'abord la vitesse MAC centrée sur les faces de la manière suivante :

$$\mathbf{U} = \mathcal{F}\left(\tilde{\mathbf{u}}\right),$$

où $\mathcal{F}$ est une fonction d'interpolation (exemple : $\mathcal{F}\left(\tilde{\mathbf{u}}_{i,j}\right) = \frac{1}{2}\left(\tilde{\mathbf{u}}_{i,j} + \tilde{\mathbf{u}}_{i+1,j}\right)$).

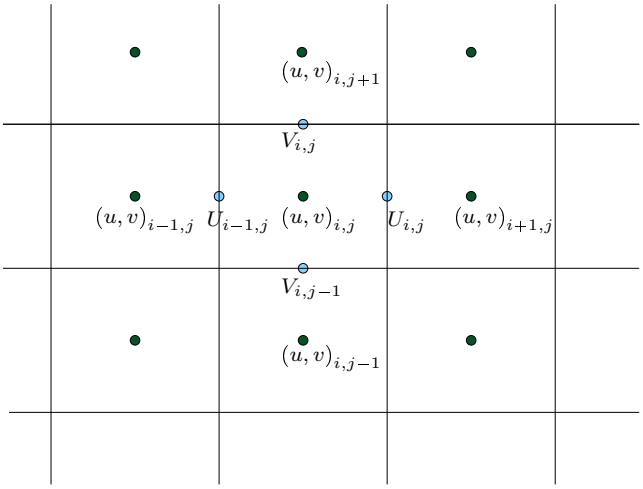


FIGURE 3.1 – Schématisation 2D de la localisation des différentes composantes des vitesses lors de la discrétisation spatiale.

Nous utilisons une méthode à pas fractionnaire de type prédicteur-correcteur pour déterminer l'évolution de la solution en temps. Nous nous basons sur les schémas de type Chorin-Temam [101][28]. On introduit donc dans un premier temps une vitesse intermédiaire prédite notée $\mathbf{u}^\star$, déterminée par la résolution de l'équation d'advection-diffusion. La discrétisation spatiale de cette équation utilise un schéma de type différences finies centrées d'ordre 2 pour les termes diffusifs tandis que le terme de convection est approché par une schéma du second ordre de type Adams-Bashforth centré sur les vitesses MAC pour plus de compacité. Cette première étape de prédiction consiste donc à déterminer $\mathbf{u}^\star$ grâce à la discrétisation suivante de l'équation (3.3) :

$$\frac{\mathbf{u}^\star - \mathbf{u}^n}{\triangle t} = -\frac{1}{2}\left(3N_i^n - N_i^{n-1}\right) - \nabla p^n + \frac{1}{2Re}\left(\triangle \mathbf{u}^n + \triangle \mathbf{u}^\star\right), \tag{3.5}$$

avec $\triangle t$ le pas de temps, $N_i = \dfrac{\delta\left(U_j u_i\right)}{\delta x_j}$ et $\triangle \mathbf{u} = \dfrac{\delta}{\delta x_j}\left(\dfrac{\delta u_i}{\delta x_j}\right)$ et $\dfrac{\delta}{\delta x}$ est discrétisé par un schéma centré du second ordre.

Cette équation est résolue en utilisant une méthode de Krylov de type GMRES [89].

On a défini le terme de convection de manière explicite à cause de sa non-linéarité, et le terme de diffusion de manière semi-implicite afin d'éliminer la contrainte de stabilité visqueuse.

Avec cette formulation, rien ne garantit que la vitesse prédite $\mathbf{u}^\star$ trouvée soit à divergence nulle. C'est pourquoi on introduit l'étape de correction.

Le principe de l'étape de correction consiste en la résolution d'une équation de Poisson dans un premier temps, puis à corriger la pression et la vitesse dans un second temps.

Faisons apparaître l'équation de Poisson.

Les équations que l'on veut à présent résoudre sont en réalité :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{\star}}{\triangle t} = -\left(\nabla p^{n+1} - \nabla p^{n}\right),$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0.$$

Prenons la divergence de l'équation (3.5), et supposons $\mathbf{u}^{n+1}$ de divergence nulle. On obtient alors :

$$\nabla \cdot \mathbf{u}^{\star} = \triangle t \nabla^2 \left(p^{n+1} - p^{n}\right).$$

Puis, on introduit la fonction $\psi = \triangle t \left(p^{n+1} - p^{n}\right)$. Par conséquent, l'équation de Poisson que nous devons résoudre est :

$$\nabla \cdot \mathbf{u}^{\star} = \triangle \psi. \tag{3.6}$$

Les conditions limites de l'équation de Poisson sont définies telles que la pression est spécifiée en un point du bord (afin d'imposer la pression loin de l'obstacle), et des conditions limites de type Neumann homogène sur le reste du domaine. Les conditions de Neumann garantissent que la vitesse ne sera pas perturbée sur les frontières du domaines pendant l'étape de correction. Par ailleurs, cette équation sera résolue grâce à l'utilisation d'une méthode de réduction cyclique[99].

À cet instant, penchons nous sur l'intérêt d'utiliser à nouveau la vitesse du schéma MAC à la place de la vitesse collocalisée.

En effet, en utilisant les vitesses définies aux centres des faces pour le calcul de la fonction ψ, le schéma est plus compacte. Cela crée un profil de pression moins oscillant. Nous utiliserons donc la vitesse définie aux centres des faces pour définir la fonction de Poisson.

Il nous faut donc actualiser les vitesses MAC tout en prenant la bonne contribution du gradient de pression, soit centré sur la face . On pose donc :

$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u} + \triangle t \left(\nabla p^{n}\right)_{cc}, \\ \tilde{\mathbf{U}} = \mathcal{F}\left(\tilde{\mathbf{u}}\right), \\ \mathbf{U}^{\star} = \tilde{\mathbf{U}} - \triangle t \left(\nabla p^{n}\right)_{fc}, \end{cases}$$

où les indices cc et fc signifie respectivement "cell centered" (i.e. centrée sur la cellule) et "face centered" (i.e. centrée sur la face).

En conséquence, l'équation (3.6) devient :

$$\nabla \cdot \mathbf{U}^{\star} = \triangle \psi,$$

où la vitesse centrée sur les faces $\mathbf{U}$ (voir Figure 3.1) est déterminée en interpolant les vitesses collocalisées $\mathbf{u}$ des cellules voisines. Cette vitesse $\mathbf{U}$ va donc nous permettre d'éliminer les oscillations de pression dues aux schémas numériques à variables collocalisées.

Enfin, les vitesses et la pression sont corrigées :

$$p^{n+1} = p^{n} + \frac{\psi}{\triangle t},$$

$$\mathbf{u^{n+1}} = \mathbf{u^\star} - (\nabla\psi)_{cc}$$
$$\mathbf{U^{n+1}} = \mathbf{U^\star} - (\nabla\psi)_{fc}\,.$$

Par ailleurs, nous utilisons un schéma d'interpolation décentré pour la détermination de la divergence dans l'équation de Poisson, le sens du décentrement dépendant de l'interpolation choisie. Enfin, notons que, dans le cas présent, la pression présente dans les équations n'est pas physique mais purement numérique. En effet, celle-ci est définie à une constante près, seule son gradient est important. En conséquence, en début de simulation nous fixerons la pression à $p = 0.5$.

En conclusion, nous utilisons donc un schéma prédicteur-correcteur avec une discrétisation spatiale d'ordre 2 mêlant approche MAC et approche collocalisée afin d'obtenir des discrétisations plus compactes. Cette méthode s'articule autour de quatre étapes :

1. Prediction :
$$\frac{\mathbf{u^\star} - \mathbf{u}^n}{\triangle t} = -\frac{1}{2}\left(3N_i^n - Ni^{n-1}\right) - \nabla p^n + \frac{1}{2Re}\left(\triangle\mathbf{u}^n + \triangle\mathbf{u^\star}\right),$$

2. Définition des vitesses MAC :
$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u^\star} + \triangle t\,(\nabla p^n)_{cc}\,, \\ \tilde{\mathbf{U}} = \mathcal{F}\,(\tilde{\mathbf{u}})\,, \\ \mathbf{U^\star} = \tilde{\mathbf{U}} - \triangle t\,(\nabla p^n)_{fc}\,, \end{cases}$$

3. Résolution de l'équation de Poisson :
$$\nabla\cdot\mathbf{U^\star} = \triangle\psi.$$

4. Correction :
$$p^{n+1} = p^n + \frac{\psi}{\triangle t},$$
$$\mathbf{u^{n+1}} = \mathbf{u^\star} - (\nabla\psi)_{cc}$$
$$\mathbf{U^{n+1}} = \mathbf{U^\star} - (\nabla\psi)_{fc}\,.$$

Maintenant que le solveur fluide a été présenté, il s'agit de présenter la modélisation de l'obstacle dans les équations. Pour cela nous allons utiliser la méthode de pénalisation (voir §2.1.1), que nous allons modifier en la fusionnant avec une approche Ghost-Cell, augmentant ainsi sa précision. Nous appellerons cette approche, *la méthode IPC*.

3.2 La méthode IPC

Dans tout le reste du chapitre, l'interface fluide-solide est définie et suivie grâce à l'utilisation de l'outil Level Set défini dans le paragraphe §1.3. De plus, nous adopterons la convention de signe présentée dans cette section.

3.2.1 Principe de la méthode

La méthode de pénalisation est du premier ordre en espace car, dans sa formulation, la frontière peut avoir n'importe quelle position dans la maille, la formule de pénalisation restera identique. Cela revient à considérer que la frontière solide coïncide tout le temps avec les noeuds des mailles créant un profil du solide "en escaliers". Or, dans les cas réalistes, la géométrie du corps est complexe. Par conséquent, l'interface fluide/solide ne coïncide pas avec les noeuds du maillage. Le principe de la méthode IPC est de corriger la valeur de pénalisation de la vitesse du solide en utilisant une approche Ghost-Cell afin d'obtenir la bonne vitesse sur l'interface immergée.

Ainsi, la principale idée dans le cas présent est de trouver la bonne correction dans la direction normale à l'interface, en utilisant les gradients de vitesse à travers l'interface. Autrement dit, considérons un cas 1D. Notons $\mathbf{u}_B$ la vitesse que nous désirons avoir à l'interface, $\bar{\mathbf{u}}$ la vitesse de pénalisation imposée sur le noeud solide en x_S et $\mathbf{u}_F$ la vitesse du point fluide situé en $x_F = x_S - \triangle x$, tout comme dans les figures 3.2 et 3.3.

La valeur de la vitesse de pénalisation $\bar{\mathbf{u}}$ doit être de telle sorte que le gradient de la vitesse à l'interface reste identique lorsqu'il est évalué de chaque côté de l'interface (voir Figure 3.2). Notons que la prolongation du gradient de vitesse à travers l'interface n'a pas de sens physique en soit, il s'agit d'un outil numérique permettant d'imposer la bonne condition à l'interface en utilisant les premiers noeuds solides.

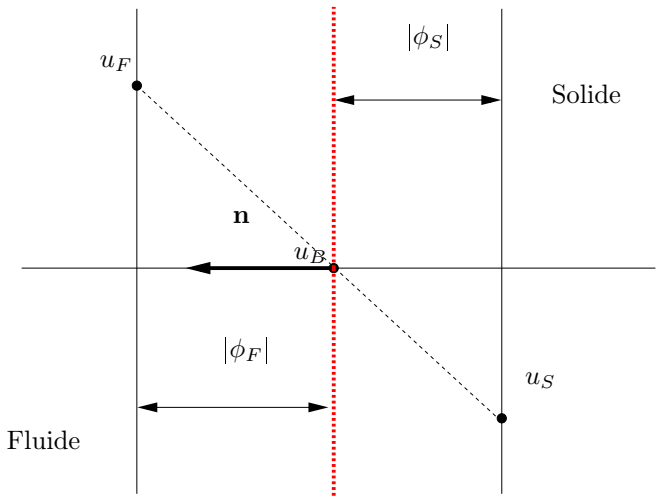


FIGURE 3.2 – Schéma 1D décrivant la méthodologie utilisée pour déterminer la vitesse des ghost cells

Mathématiquement, cela signifie :

$$\frac{\bar{\mathbf{u}} - \mathbf{u}_B}{|\phi_S|} = \frac{\mathbf{u}_B - \mathbf{u}_F}{|\phi_F|},$$

où $|\phi_S|$ et $|\phi_F|$ sont les valeurs absolues de la fonction Level Set en $\mathbf{x}_S$ et $\mathbf{x}_F$ respecti-

vement, soit, leur distance à l'interface. Autrement dit,

$$\bar{\mathbf{u}} = \mathbf{u}_B + |\phi_S| \left(\frac{\mathbf{u}_B - \mathbf{u}_F}{|\phi_F|} \right),$$

ce qui nous permet d'établir l'expression générale de la vitesse "corrigée" :

$$\bar{\mathbf{u}} = \mathbf{u}_B - |\phi| \left(\frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \Bigg|_{\phi=0},$$

avec $\mathbf{n}$ le vecteur unitaire de la normale extérieure à l'interface.

Finalement, étant donné que ϕ est négative dans le solide, la forme finale de la correction IPC est :

$$\bar{\mathbf{u}} = \mathbf{u}_B + \phi \left(\frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \Bigg|_{\phi=0}. \tag{3.7}$$

Notons enfin que seuls les "Ghost nodes" subissent cette correction ; on applique une pénalisation classique à tous les autres points.

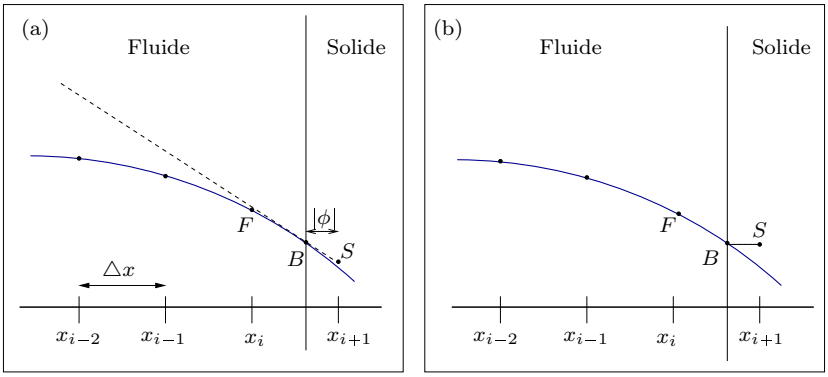


FIGURE 3.3 – Schéma 1D représentant les deux types de méthode : (a) méthode IPC et (b) méthode de pénalisation classique, avec S le point solide soumis à la pénalisation, B le point sur l'interface et F le point fluide voisin.

Maintenant, considérons un cas général (voir Figure 3.4 pour visualiser un exemple 2D).

Dans le but de déterminer $\left(\frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \Big|_{\phi=0}$, nous devons déterminer la position et la vitesse de tous les points symétriques aux ghost points par rapport à l'interface dans la direction normale.

La détermination de la position des points symétriques peut s'effectuer aisément grâce à l'outil Level Set. En effet, comme il l'a été vu dans §1.3, le vecteur normal unitaire extérieur à l'interface peut être calculé sur la simple connaissance de la fonction Level Set associée au solide. De plus, connaissant la direction, nous savons que le point sur l'interface immergée se situe à une distance de $|\phi|$ du ghost point, par conséquent, le point symétrique au ghost point se situera à une distance de $2\,|\phi|$. On s'aperçoit ici de l'importance d'une définition précise de $\nabla\phi$: la définition du vecteur normal,

donc du point symétrique et par conséquent de la correction de la vitesse de pénalisation en sont complètement tributaires. $\nabla\phi$ doit donc être calculé de manière très précise.

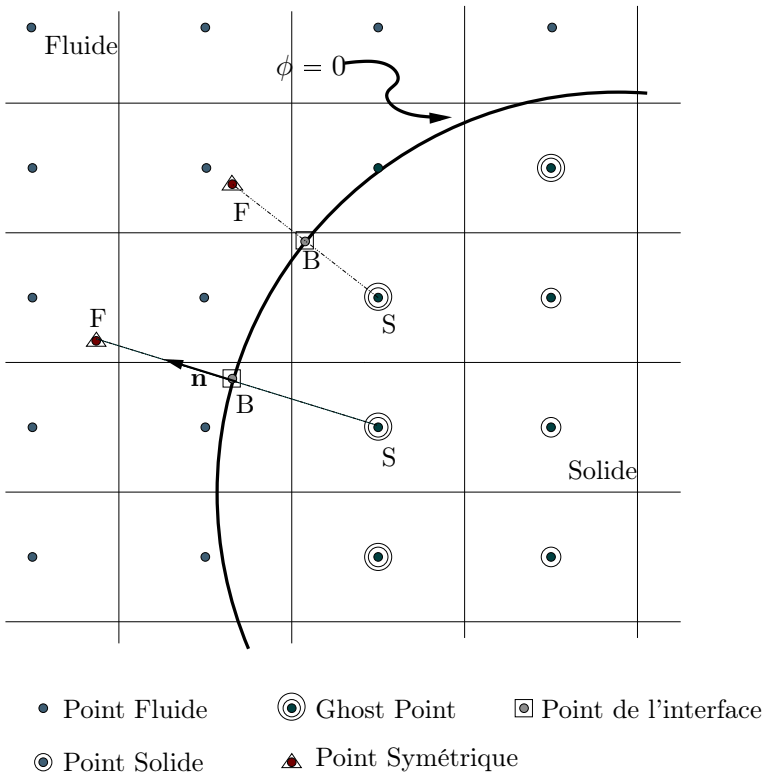


FIGURE 3.4 – Schéma 2D décrivant les éléments utilisés par le solveur pour déterminer la correction. La correction est appliquée au ghost points en utilisant les points symétriques. La vitesse du point symétrique est basée sur une interpolation utilisant les quatre voisins.

Naturellement, le point symétrique ne coïncide pas avec un noeud de la grille. Nous devons donc interpôler sa vitesse à partir de ses huit (quatre en 2D) plus proches voisins. C'est dans cette optique que nous avons utilisé des polynômes de Lagrange du second ordre. La procédure de détermination du point symétrique est la suivante :

Considérons un cas 3D. Notons (x, y, z) les trois composantes de la position du point symétrique. De même on note $\{x_1, x_2, y_1, y_2, z_1, z_2\}$ les plus petites et grandes composantes de position, pour les huit noeuds voisins les plus proches tels que $(x, y, z) \in [x_1, x_2] \times [y_1, y_2] \times [z_1, z_2]$. Pour plus de légèreté dans les notations, huit points sont introduits : M, N, P, Q, M', N', P' et Q', définis par (voir Figure 3.5) :

$$\begin{cases} \mathbf{x_M} = (x_1, y_1, z_1)\,, \\ \mathbf{x_N} = (x_2, y_1, z_1)\,, \\ \mathbf{x_P} = (x_2, y_2, z_1)\,, \\ \mathbf{x_Q} = (x_1, y_2, z_1)\,. \end{cases} \qquad \begin{cases} \mathbf{x_{M'}} = (x_1, y_1, z_2)\,, \\ \mathbf{x_{N'}} = (x_2, y_1, z_2)\,, \\ \mathbf{x_{P'}} = (x_2, y_2, z_2)\,, \\ \mathbf{x_{Q'}} = (x_1, y_2, z_2)\,. \end{cases}$$

$L_M, L_N, L_P, L_Q, L_{M'}, L_{N'}, L_{P'}$ et $L_{Q'}$ sont leur polynôme de Lagrange respectif correspondant à l'interpolation :

$$\mathbf{u_F} = \sum_{i=M}^{Q'} \mathbf{u_i} L_i\left(\mathbf{x_F}\right),$$

et $\{L_i\left(\mathbf{x_F}\right)\}_i$ prend la forme

$$\begin{cases} L_M\left(\mathbf{x_F}\right) = \dfrac{(x-x_2)(y-y_2)(z-z_2)}{(x_1-x_2)(y_1-y_2)(z_1-z_2)}, \\ L_N\left(\mathbf{x_F}\right) = \dfrac{(x-x_1)(y-y_2)(z-z_2)}{(x_2-x_1)(y_1-y_2)(z_1-z_2)}, \\ L_P\left(\mathbf{x_F}\right) = \dfrac{(x-x_1)(y-y_1)(z-z_2)}{(x_2-x_1)(y_2-y_1)(z_1-z_2)}, \\ L_Q\left(\mathbf{x_F}\right) = \dfrac{(x-x_2)(y-y_1)(z-z_2)}{(x_1-x_2)(y_2-y_1)(z_1-z_2)}. \end{cases} \quad \begin{cases} L_{M'}\left(\mathbf{x_F}\right) = \dfrac{(x-x_2)(y-y_2)(z-z_1)}{(x_1-x_2)(y_1-y_2)(z_2-z_1)}, \\ L_{N'}\left(\mathbf{x_F}\right) = \dfrac{(x-x_1)(y-y_2)(z-z_1)}{(x_2-x_1)(y_1-y_2)(z_2-z_1)}, \\ L_{P'}\left(\mathbf{x_F}\right) = \dfrac{(x-x_1)(y-y_1)(z-z_1)}{(x_2-x_1)(y_2-y_1)(z_2-z_1)}, \\ L_{Q'}\left(\mathbf{x_F}\right) = \dfrac{(x-x_2)(y-y_1)(z-z_1)}{(x_1-x_2)(y_2-y_1)(z_2-z_1)}. \end{cases}$$

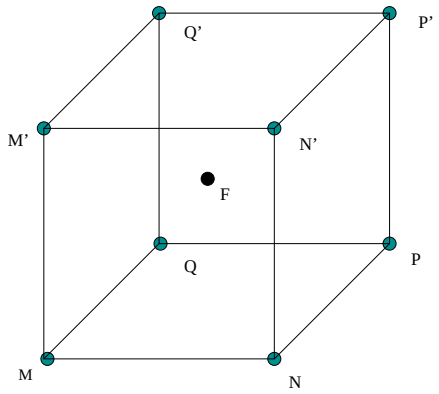


FIGURE 3.5 – Représentation 3D de la position des points employés pour l'interpolation sur la vitesse du point symétrique

Enfin, notons que le point symétrique peut se situer dans une maille possédant un ghost point au moins comme noeud. Cela pourrait poser problème pour calculer la précision spatiale si l'erreur en temps devient comparable avec l'erreur en espace. Pour éviter ce genre de cas, nous utiliserons un pas de temps très petit devant le pas d'espace.

3.2.2 Application aux équations de Navier Stokes

L'application de la méthode IPC aux équations de Navier Stokes est "presque" en tout point similaire à la méthode classique de pénalisation. Tout d'abord, les équations de Navier Stokes s'écrivent à présent :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\nabla p + \frac{1}{Re}\triangle \mathbf{u} + \frac{\chi_{gc}}{K}\left(\mathbf{u} - \bar{\mathbf{u}}\left(\mathbf{x}, t\right)\right), \tag{3.8}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3.9}$$

où $\bar{\mathbf{u}}$ est déterminée par la procédure décrite précédemment et χ_{gc} est la fonction indicatrice des ghost cells, autrement dit,

$$\chi_{gc}\left(i,j\right) = \begin{cases} 1 & \text{si } (i,j) \text{ est une ghost cell,} \\ 0 & \text{sinon.} \end{cases}$$

La différence de résolution dans le cas présent entre la méthode de pénalisation classique et la méthode IPC se situe au niveau des étapes de résolution numérique des équations de Navier Stokes. En effet, la méthode IPC introduit une vitesse inexacte sur les ghost points. L'effet sera bénéfique pour les calculs des termes de diffusion et de convection, mais cela deviendra gênant pour le calcul de la divergence dans l'étape de poisson. En effet, les valeurs de divergence calculées dans le domaine intérieur au solide deviennent fausses.

Par conséquent, pour éviter l'introduction d'une erreur dans l'étape de résolution de l'équation de Poisson, on introduit une pénalisation classique entre l'étape de prédiction et de résolution de l'équation de Poisson, et enfin, on effectue l'étape IPC après l'étape de correction. De plus, nous introduisons également cette étape de pénalisation classique car celle-ci applique les bonnes conditions de pression à l'interface, sans aucun traitement supplémentaire. En effet, au niveau de l'interface immergée, la condition de pression suivante doit être satisfaite :

$$\frac{\partial p}{\partial \mathbf{n}} = 0. \tag{3.10}$$

Lorsque l'on n'utilise pas la méthode de pénalisation, la condition (3.10) est imposée durant l'étape de Poisson et doit être mise à jour à chaque itération, ce qui peut être très coûteux, surtout lorsque l'on utilise également un préconditionneur. Grâce à la méthode de pénalisation, cette mise à jour devient inutile étant donné que les conditions de pression sont imposées à la surface de manière implicite [3]. La figure 3.6 illustre cette propriété dans deux situations : le cas d'un cylindre fixe et d'un cylindre mobile. On s'aperçoit que les deux profils sont très similaires bien que dans le second cas, du bruit apparaisse à cause du mouvement du corps.

Finalement, l'ensemble des étapes de résolution devient :

1. Prediction avec pénalisation classique :

$$\frac{\mathbf{u}^{\star} - \mathbf{u}^{n}}{\triangle t} = -\frac{1}{2}\left(3N_i^n - Ni^{n-1}\right) - \nabla p^n + \frac{1}{2Re}\left(\triangle \mathbf{u}^n + \triangle \mathbf{u}^{\star}\right) + \frac{\chi_s}{K}\left(\mathbf{u_B} - \mathbf{u}^{\star}\right),$$

2. Définition des vitesses aux centres des faces :

$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u}^{\star} + \triangle t\left(\nabla p^n\right)_{cc}, \\ \tilde{\mathbf{U}} = \mathcal{F}\left(\tilde{\mathbf{u}}\right), \\ \mathbf{U}^{\star} = \tilde{\mathbf{U}} - \triangle t\left(\nabla p^n\right)_{fc}, \end{cases}$$

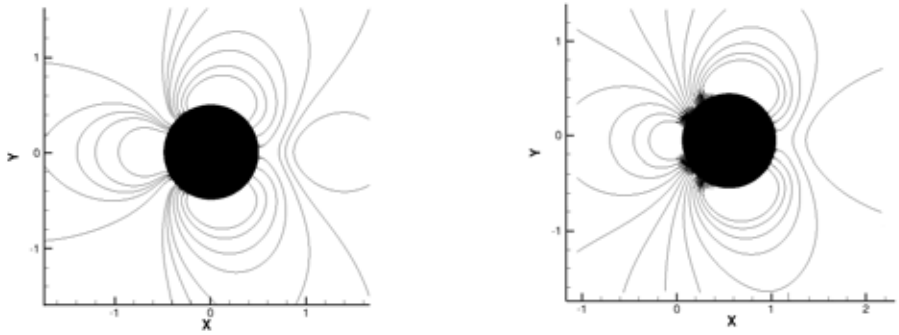


FIGURE 3.6 – Distribution des iso-contours de pression dans le cas d'un cylindre fixe (à gauche) et d'un cylindre mobile (à droite). Les deux images correspondent à une solution pour une grille de taille (360×240).

3. Résolution de l'équation de Poisson :

$$\nabla \cdot \mathbf{U}^{\star} = \triangle \psi,$$

4. Correction :

$$p^{n+1} = p^n + \frac{\psi}{\triangle t},$$

$$\tilde{\mathbf{u}}^{\mathbf{n+1}} = \mathbf{u}^{\star} - (\nabla \psi)_{cc},$$

$$\mathbf{U}^{\mathbf{n+1}} = \mathbf{U}^{\star} - (\nabla \psi)_{fc},$$

5. IPC :

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} + \frac{\chi_{gc}}{K}\left(\mathbf{u}^{n+1} - \bar{\mathbf{u}}\left(\mathbf{x}, t\right)\right).$$

Remarque : le lien existant entre la fonction Level Set à l'interface du solide et sa fonction caractéristique est le suivant :

$$\chi_S = 1 - H\left(\phi\right)$$

où H désigne la fonction de Heaviside.

3.2.3 Cas d'un solide en mouvement

Lorsque l'on considère un corps en mouvement, on peut rencontrer des problèmes dans les cellules nouvellement fluides, autrement dit des cellules qui étaient solides au temps t^n et sont devenues fluides au temps t^{n+1} dû au déplacement de l'interface [70][110][109]. Dans de telles situations, une éventuelle solution peut être d'évaluer la valeur de la vitesse dans ces nouvelles cellules en utilisant une interpolation sur ses cellules fluides voisines. Ce fut la méthode utilisée par Ye *et al.* dans [122].

Dans notre cas, grâce à nos différentes étapes, une autre approche est possible. Le point clé étant le moment où l'on transportera l'interface.

Étant donné que l'on utilise la méthode du splitting, *a priori*, on pourrait déplacer la fonction Level Set à n'importe quel moment durant notre résolution. Toutefois, étant donné que nous avons une étape de correction des ghost cells, si la fonction Level Set est transportée en dernière étape, les nouveaux points dont les valeurs auront été corrigées seront, soit dans le fluide (et leur valeur sera erronée), soit solides mais ne seront plus des ghost points (et leur valeur sera encore une fois fausse puisqu'ils devraient être soumis à une pénalisation classique). Par conséquent, l'étape de transport de la fonction Level Set doit être effectuée avant l'étape IPC.

Par ailleurs, nous constatons également le troisième intérêt de l'étape de pénalisation classique. En effet, supposons que nous n'effectuons pas cette étape de pénalisation classique. Dès lors, quel que soit l'autre moment où nous décidons de déplacer l'interface, nous sommes confrontés au même problème : des points solides ayant subis l'IPC à l'itération précédente, deviendront fluides et auront une vitesse erronée. Or, dans notre cas, si nous déplaçons la fonction Level Set après l'étape de pénalisation classique, les points nouvellement fluides auront une valeur correcte.

En conséquence, nous avons décidé de déplacer l'interface après l'étape de correction des vitesses et pression.

Finalement, l'ensemble final des étapes de l'algorithme de résolution est le suivant :

1. Prediction :
$$\frac{\mathbf{u}^\star - \mathbf{u}^n}{\triangle t} = -\frac{1}{2}\left(3N_i^n - Ni^{n-1}\right) - \nabla p^n + \frac{1}{2Re}\left(\triangle \mathbf{u}^n + \triangle \mathbf{u}^\star\right) + \frac{\chi_s}{K}\left(\mathbf{u_B} - \mathbf{u}^\star\right),$$

2. Définition des vitesses aux centres des faces :
$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u}^\star + \triangle t \left(\nabla p^n\right)_{cc}, \\ \tilde{\mathbf{U}} = \mathcal{F}\left(\tilde{\mathbf{u}}\right), \\ \mathbf{U}^\star = \tilde{\mathbf{U}} - \triangle t \left(\nabla p^n\right)_{fc}, \end{cases}$$

3. Résolution de l'équation de Poisson :
$$\nabla \cdot \mathbf{U}^\star = \triangle \psi,$$

4. Correction :
$$p^{n+1} = p^n + \frac{\psi}{\triangle t},$$
$$\tilde{\mathbf{u}}^{\mathbf{n+1}} = \mathbf{u}^\star - \left(\nabla \psi\right)_{cc},$$
$$\mathbf{U}^{\mathbf{n+1}} = \mathbf{U}^\star - \left(\nabla \psi\right)_{fc},$$

5. Transport de la fonction Level Set :
$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0,$$

6. IPC :
$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} + \frac{\chi_{gc}}{K}\left(\mathbf{u}^{n+1} - \bar{\mathbf{u}}\left(\mathbf{x}, t\right)\right).$$

Remarque : dans cette thèse, au cours de nos études, nous connaissons toujours exactement la position de l'interface fluide-structure. Nous calculons donc manuellement la fonction distance signée en déplaçant l'isoligne zéro à la vitesse $\mathbf{u}$, puis déterminons l'ensemble des contour de la fonction Level Set dans le domaine d'étude. Nous ne transportons donc jamais l'ensemble de la fonction Level Set.

3.3 Résultats Numériques

Dans cette section nous évaluerons dans un premier temps la précision de la méthode, puis nous comparerons la solution fournie par le solveur à différents cas test issus de la littérature, dans le cas d'un ou plusieurs corps solide(s), fixe(s) puis nous finirons par des simulations d'un corps mobile.

3.3.1 Étude de la convergence

Les schémas de différences finis utilisés lors de la discrétisation spatiales sont globalement du second ordre. La correction que nous appliquons dans l'étape de pénalisation fut introduite dans le but de maintenir ce second ordre lorsque l'on impose une condition de bord sur la vitesse. En conséquence, nous nous attendons à une précision d'ordre deux globalement.

Dans le cas présent, nous testons le solveur dans le cas bidimensionnel d'un écoulement en présence d'un cylindre circulaire fixe avec un nombre de Reynolds $Re = 45$. Le domaine d'étude est de taille $6D \times 4D$ avec D le diamètre du cylindre. Son centre est situé en $(0, 0)$.

L'étude de convergence consiste à comparer les erreurs pour différentes grilles, entre la solution fournie par le solveur et la solution analytique associée au problème considéré. Néanmoins, comme il n'existe pas de solution analytique pour un tel cas test, nous amalgamerons une solution évaluée sur une grille raffinée avec la solution analytique. La solution raffinée correspondra à la solution évaluée pour une grille de taille (1800×1200). Cette solution est comparée à cinq autres grilles correspondant aux maillages de tailles suivantes : (600×400), (360×240), (300×200), (225×150) et (180×120). Le pas de temps choisi pour l'ensemble des simulations est $10^{-5} D/U_\infty$, respectant ainsi la condition CFL :

$$\left(\frac{|u|}{\triangle x} + \frac{|v|}{\triangle y} \right) \triangle t \leqslant CFL.$$

Les différentes solutions sont comparées après 5.10^4 itérations, soit au temps $t = 0.5$. La répartition des amplitudes d'erreurs en vitesse et pression pour la grille (360×240) est donnée en Figure 3.7. Cette figure montre que l'erreur est principalement concentrée autour de la frontière immergée. Enfin, la figure 3.8 montre que nous obtenons du second ordre globalement, mesuré avec la norme L^2, sur la vitesse et de l'ordre 1.5 sur la pression. En outre, nous obtenons également de l'ordre 2 en vitesse localement, évalué avec la norme L_∞, et de l'ordre 0.5 en pression. D'après la figure 3.7, l'erreur en pression est d'amplitude dix fois supérieure à celle de la vitesse et reste concentrée autour de l'interface. Cela peut provenir de notre correction de la vitesse en étape finale qui crée un champ de vitesse

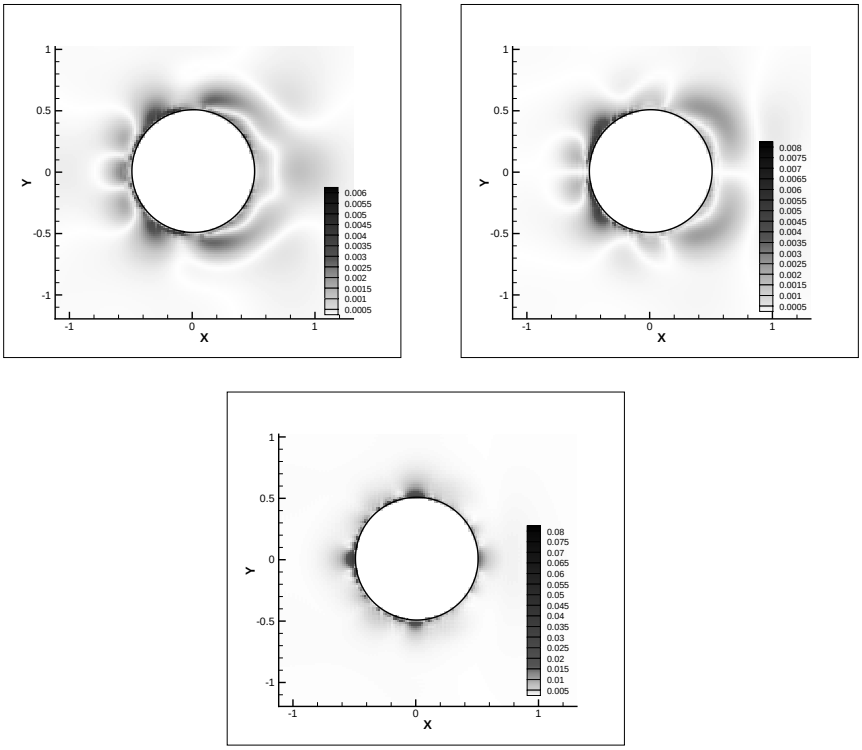


FIGURE 3.7 – (en haut, à gauche) : Répartition de l'erreur en u, (en haut à droite) : Répartition de l'erreur en v, (en bas) : Répartition de l'erreur en pression, toutes les images correspondent à la solution numérique pour la grille de taille (360×240).

qui n'est plus à divergence nulle. Par conséquent, les corrections en vitesse et pression seraient erronées.

3.3.2 Cas d'un cylindre, fixe ou mobile

Considérons un écoulement 2D sur un cylindre circulaire. Nous voulons valider notre méthode pour des cas tests couvrant une large gamme de nombres de Reynolds [1].

On sait qu'un écoulement présentant un Reynolds allant jusque 49 approximativement est stable et possède une zone de recirculation symétrique par rapport à la ligne centrale du sillage. Pour des nombres de Reynolds plus élevés allant jusque 190, l'écoulement est inconditionnellement stable et crée un phénomène périodique nommé les allées de Von Kàrmàn. Après le seuil $Re = 190$, l'écoulement présente un régime transitoire provoquant des instabilités tridimentionnelles qui conduiront à la turbulence pour de plus forts Reynolds.

Nous avons effectué des simulations pour différents nombres de Reynolds : 40, 100, 200, 300 et 550, couvrant ainsi toutes les situations sus-mentionnées.

1. Les différents Reynolds correspondent à des différentes valeurs de viscosité. On prendra toujours $D = 1$ et $U_\infty = 1$.

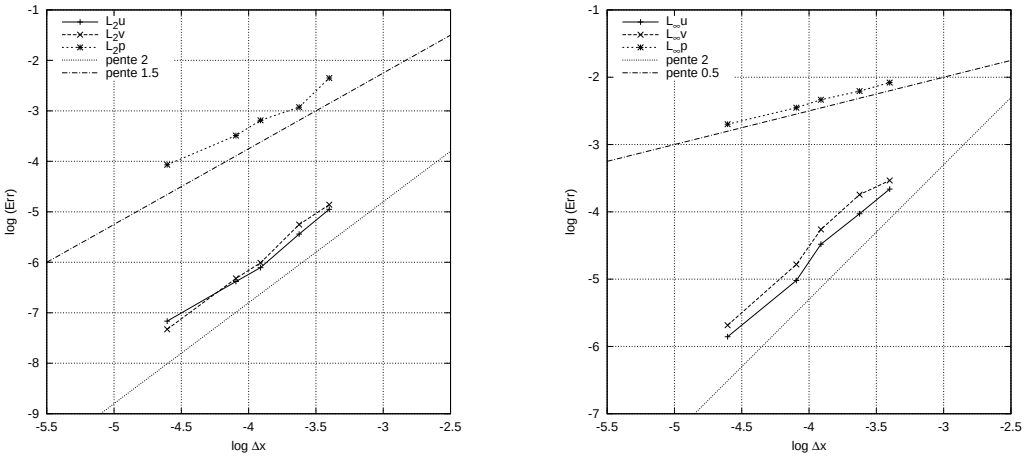


FIGURE 3.8 – Courbes d'ordre en norme L_2 (à gauche) et L_∞ (à droite) pour les composantes de vitesse u et v et pour la pression, en fonction de la taille de la grille.

3.3.2.1 Évaluation des forces

Dans le but de calculer les coefficients aérodynamiques $\mathbf{C} = \mathbf{F} / \left(\rho_0 U_\infty^2 \frac{1}{2} D \right)$, nous utilisons la formulation des forces du fluide sur un corps rigide suivante :

$$\mathbf{F} = \oint_{\partial\Omega_S} \mathbf{n} \left[-pI + \frac{1}{Re} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] \mathrm{d}s, \tag{3.11}$$

où $\partial\Omega_S$ représente la frontière du corps solide et $\mathbf{n}$ le vecteur normal unitaire extérieur au solide.

Afin de pouvoir évaluer la pression et les gradients de vitesse à l'interface, nous utiliserons une interpolation $\mathbb{Q}_1$ sur la pression et la vitesse. Nous avons choisi cette interpolation car elle nous permettra d'obtenir directement les gradients de vitesse.

L'interpolation $\mathbb{Q}_1$ consiste à approcher la variable d'écoulement φ (ici la pression et la vitesse) par l'interpolation quadratique présentée en §2.1.2.1. En l'occurrence, après avoir repéré lagrangiennement le solide grâce à des marqueurs, pour chaque marqueur, on repère les quatre noeuds du maillage qui lui sont les plus proches. Puis, on détermine la valeur de la variable d'écoulement φ en ce point par la formulation

$$\varphi(x, y) = c_1 xy + c_2 x + c_3 y + c_4,$$

avec (x, y) la position du marqueur courant. Les quatre coefficients c_i seront déterminés grâce aux valeurs de φ aux différents noeuds voisins identifiés préalablement :

$$\{\underline{C}\} = [\mathcal{V}]^{-1} \{\underline{\varphi}\},$$

où $\begin{cases} \{\underline{C}\}^T = (c_1, ..., c_4) \\ \quad \text{le vecteur contenant les quatre coefficients inconnus,} \\ \{\underline{\varphi}\}^T = (\varphi_1, ..., \varphi_4) \\ \quad \text{les valeurs de la variable d'écoulement sur les quatre noeuds,} \\ [\mathcal{V}] \text{ est la matrice de Vandermonde.} \end{cases}$

$[\mathcal{V}]$ s'écrit donc :

$$[\mathcal{V}] = \begin{pmatrix} xy|_1 & x|_1 & y|_1 & 1 \\ xy|_2 & x|_2 & y|_2 & 1 \\ xy|_3 & x|_3 & y|_3 & 1 \\ xy|_4 & x|_4 & y|_4 & 1 \end{pmatrix}.$$

De cette manière, on peut déterminer la pression, et en dérivant cette expression par x ou y, on peut déterminer les composantes du gradient de vitesse.

Pour tous les tests suivants, le domaine de calcul est de dimension $[-8; 16] \times [-8; 8]$. Le pas d'espace est fixé à $\triangle x = 0.01$ et le pas en temps est choisi de telle sorte qu'il respecte la condition CFL.

Les coefficients de trainée et de portance sont respectivement définis comme : $C_D = \mathbf{C}.\mathbf{e}_x$ et $C_L = \mathbf{C}.\mathbf{e}_y$.

3.3.2.2 Cas du cylindre fixe

Le premier cas que nous considérons est un écoulement à Reynolds 550, en présence d'un cylindre circulaire fixe avec $U_\infty = 1$. L'évolution temporelle de la trainée est donnée en figure 3.9. Les résultats trouvés sont en bon accord avec les résultats numériques présentés dans [81].

Dans un second temps, nous considérons toujours un cylindre fixe mais deux écoulements distincts[2] : le premier est à $Re = 40$ et le second à $Re = 200$. Les évolutions en temps des coefficients de trainée et de portance sont présentées en figure 3.10.

Nous avons effectué la même simulation pour d'autres nombres de Reynolds. Les différents résultats sont rassemblés dans la Table 3.1 et concordent avec les études précédentes.

2. Les différents Reynolds correspondent à des différentes valeurs de viscosité. On prendra toujours $D = 1$ et $U_\infty = 1$.

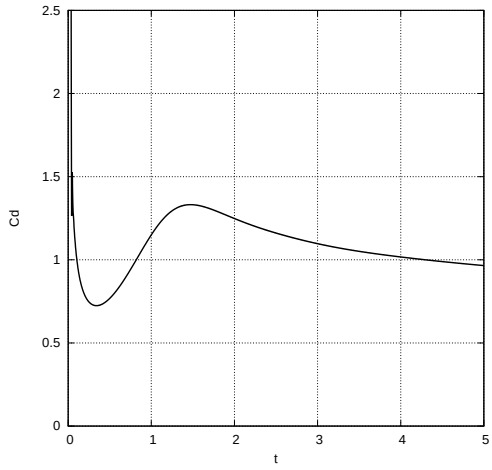


FIGURE 3.9 – Coefficient de trainée pour un cylindre fixe à $Re = 550$.

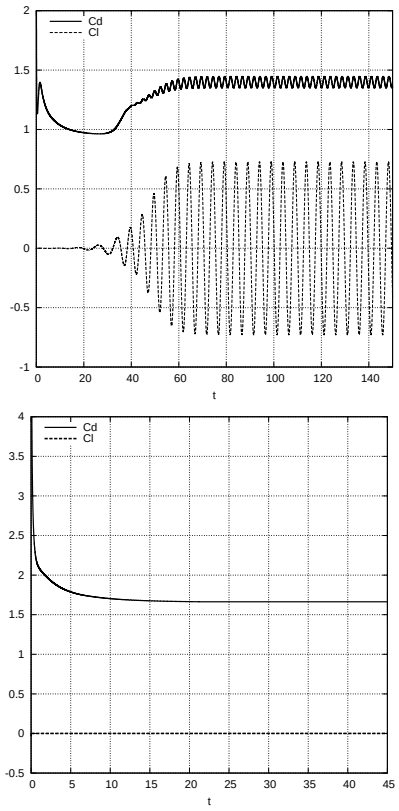


FIGURE 3.10 – Coefficients de trainée et de portance à $Re = 200$ (en haut) et à $Re = 40$ (en bas).

3.3.2.3 Cas du cylindre instantanément mis en mouvement

Afin d'évaluer la capacité de notre méthode à gérer les interfaces mobiles, nous considérons tout d'abord le même cas que précédemment à $Re = 550$ mais en considérant un

Re	40	100	200	300
Présente étude	1.66	1.41	1.40	1.42
Wieselsberger [118]	1.70	-	-	1.22
Mittal [70]	1.53	1.35	-	1.36
Hendersen [50]	1.54	1.35	1.34	1.37
Braza [19]	-	1.36	1.40	-
Bergmann [12]	1.68	1.41	1.39	-

TABLE 3.1 – Comparaison du coefficient de trainée avec des études précédentes.

fluide initialement au repos et un cylindre mis en mouvement instantanément à la vitesse horizontale $u_S = -1$. À travers la figure 3.11, nous constatons que notre méthode présente de bons résultats puisque le profil de trainée est semblable à celui de la Figure 3.9. De plus, les oscillations (décroissantes quand $\triangle x \to 0$) présentes sur l'évolution du coefficient trainée dans le cas du cylindre mobile sont de plus faible amplitude lorsque la méthode IPC est utilisée en comparaison avec l'emploi de la pénalisation classique. En effet, l'amplitude des oscillations obtenues en utilisant la méthode IPC et mesurées entre les temps t=3 et t=5, est 4 fois inférieure à celles établies en utilisant la pénalisation classique.

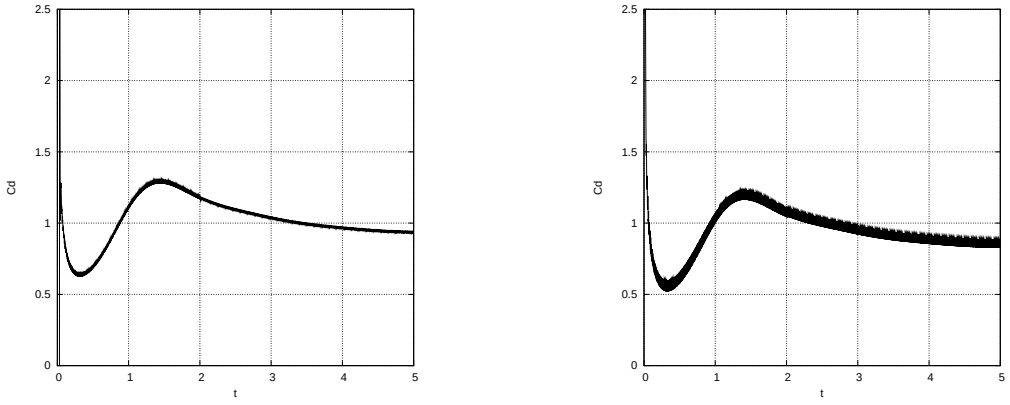


FIGURE 3.11 – Comparaison des coefficients de trainée pour un cylindre mobile à $Re = 550$, en utilisant la méthode IPC (en haut) et la pénalisation classique (en bas).

Ensuite, nous avons décidé de comparer les évolutions du coefficient de trainée pour $Re = 40$ et $Re = 200$, avec celles données par Koumoutsakos *et al.* dans [61], qui nous ont permis de comparer nos résultats aux leurs.

Notons que dans [61], le temps est adimensionné en utilisant le rayon du cylindre : $T_0 = \frac{\frac{1}{2}D}{U_\infty}$, tandis que nous utilisons le diamètre : $t_0 = \frac{D}{U_\infty}$, ce qui explique le facteur 2 entre leur temps adimensionné et le nôtre. Les comparaisons ci-après seront adaptées à notre temps adimensionné.

Les simulations se déroulent jusqu'au temps $t = 0.25$ et la figure 3.12 montre que nos résultats sont en très bon accord avec ceux présentés par Koumoutsakos *et al.*

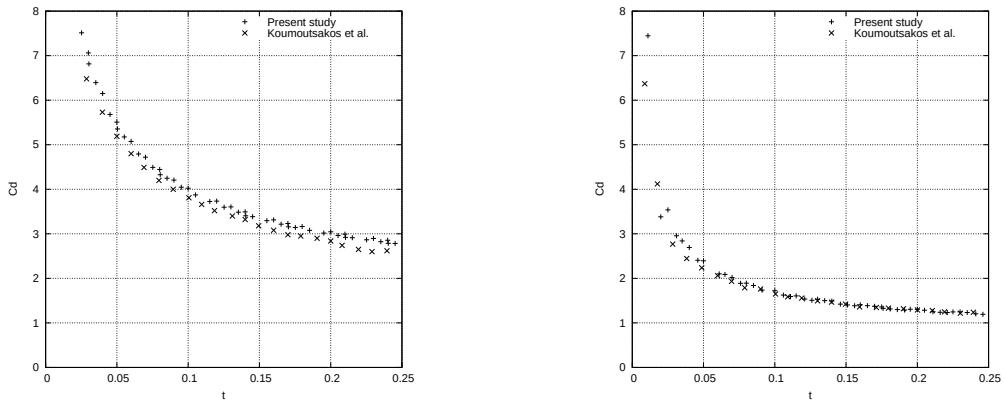


FIGURE 3.12 – Comparaison des coefficients de trainée pour un cylindre mis en mouvement instantanément, à $Re = 40$ (en haut) et $Re = 200$ (en bas) , avec l'étude de Koumoutsakos *et al.* [61].

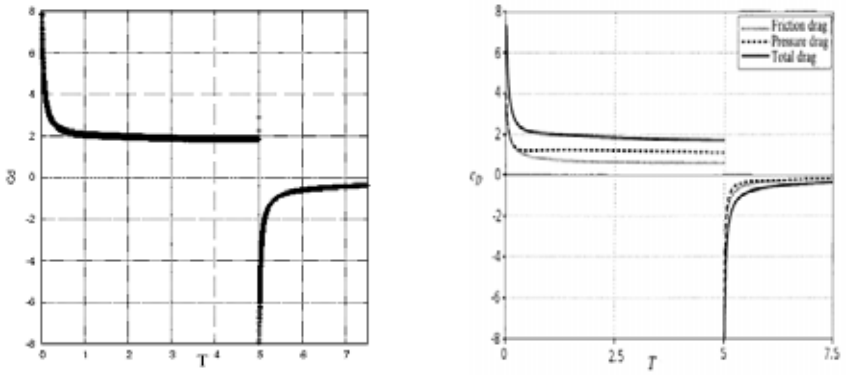


FIGURE 3.13 – Comparaison des coefficients de trainée pour un cylindre mis en mouvement, instantanément arrêté au temps $t = 5$ à $Re = 40$: notre méthode à gauche, celle de Koumoutsakos *et al.* à droite.

3.3.2.4 Cas du cylindre instantanément mis en mouvement puis arrêté

Nous avons également effectué des simulations 2D d'un cylindre instantanément mis en mouvement puis arrêté, pour différents nombres de Reynolds. Ces cas tests sont également présentés par Koumoutsakos *et al.* dans [61].

Nous nous plaçons tout d'abord à $Re = 40$. Nous considérons un cylindre initialement mis en mouvement puis arrêté au temps $t = 5$. La figure 3.13 montre une comparaison des évolutions temporelles du coefficient de trainée, où nos résultats concordent avec ceux de Koumoutsakos *et al.* La figure 3.14 représente le champ de vorticité calculé avec notre méthode IPC, qui correspond également avec les profils exposés dans [61] et représentés en figure 3.15.

Pour finir, nous avons effectué une dernière simulation. À $Re = 550$, un cylindre circulaire est mis en mouvement puis instantanément arrêté au temps $t = 2.5$. La figure 3.16 compare les évolutions en temps des coefficients de trainée. Enfin, les figures 3.17

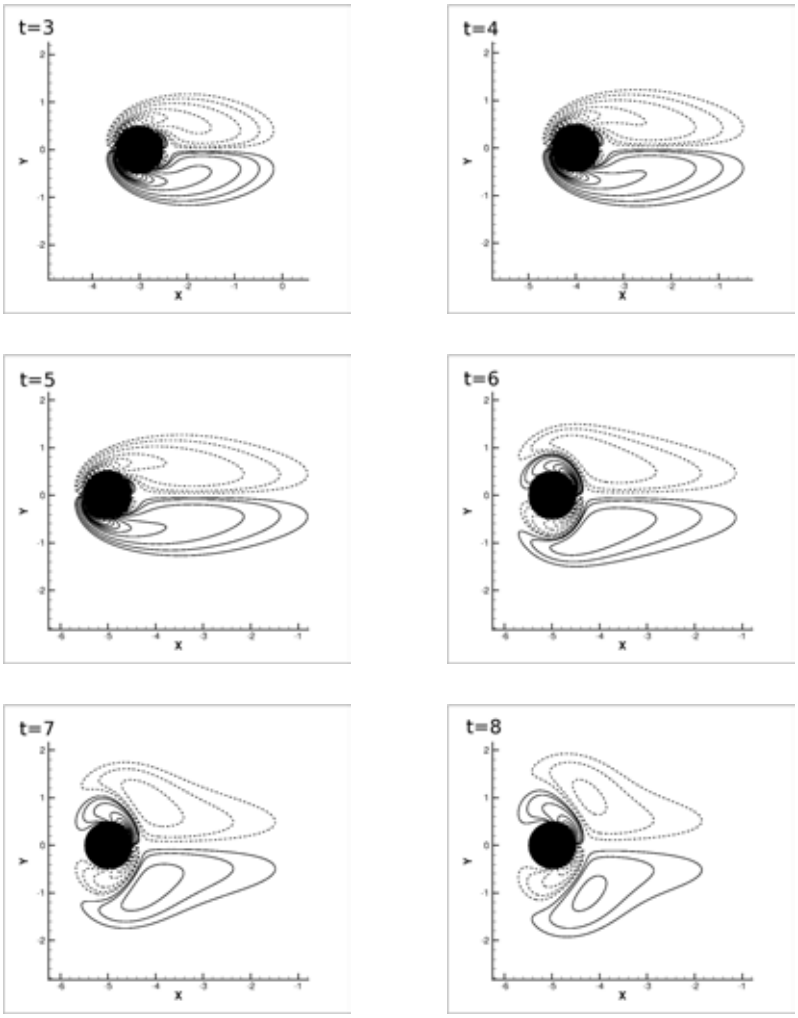


FIGURE 3.14 – Isolignes de vorticité calculé avec notre méthode pour un cylindre mis en mouvement instantanément arrêté au temps $t = 5$ à $Re = 40$.

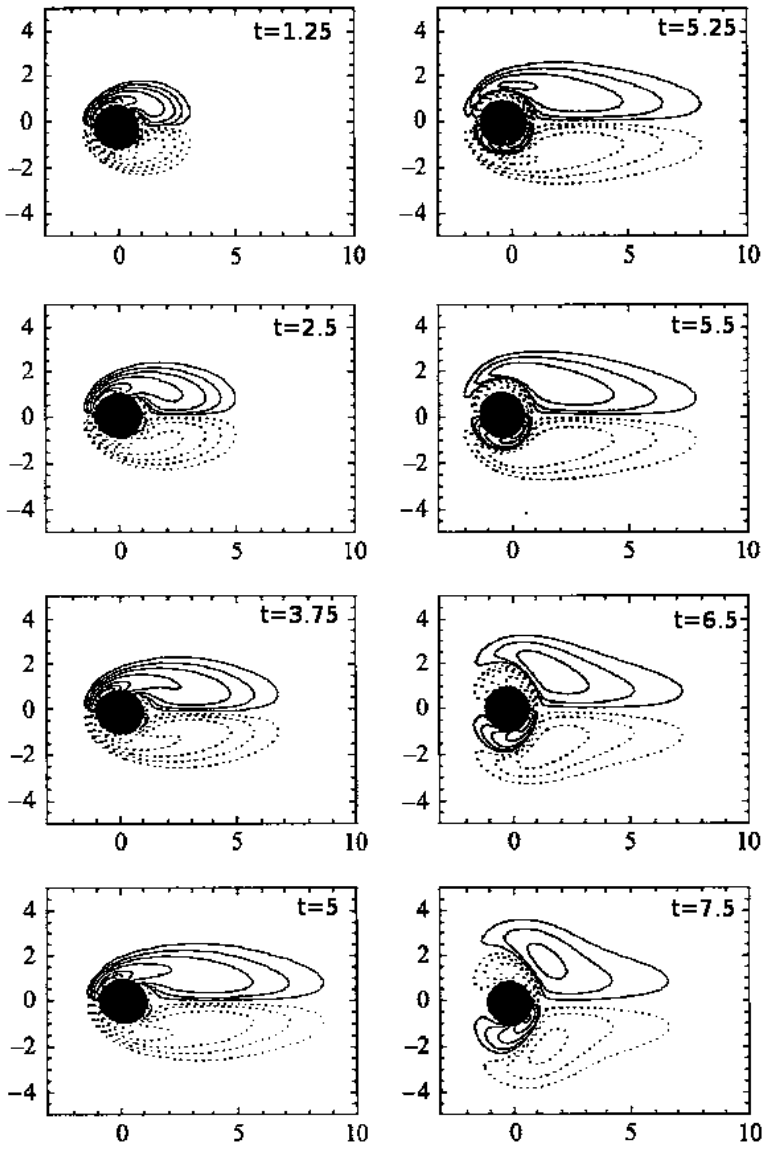


FIGURE 3.15 – Isolignes de vorticité présenté par Koumoutsakos *et al.* dans [61], pour un cylindre mis en mouvement instantanément arrêté au temps $t = 5$ à $Re = 40$.

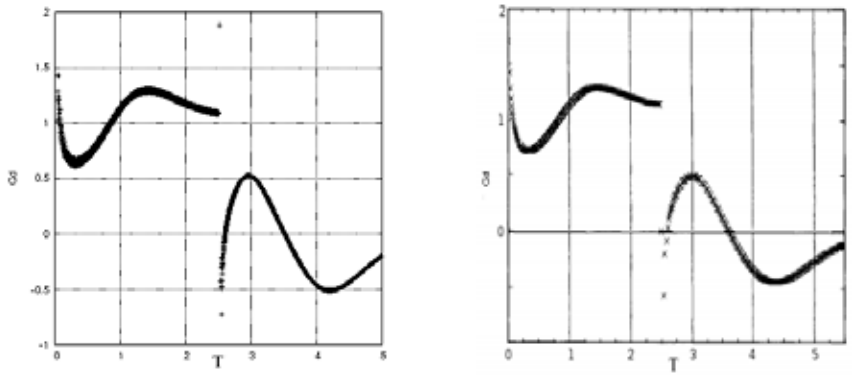


FIGURE 3.16 – Comparaison des coefficients de trainée pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$: notre méthode à gauche, celle de Koumoutsakos *et al.* à droite [61].

et 3.18 présentent les champs de vorticité obtenus avec notre méthode et la méthode de Koumoutsakos *et al.* respectivement. Tous les résultats sont concordants.

3.3.3 Cas de deux cylindres fixes

Cherchant maintenant à valider notre méthode dans le cas de plusieurs corps solides, nous nous sommes placés dans la configuration décrite ci-après.

Considérons deux cylindres fixes circulaires, $\mathcal{C}_1$ et $\mathcal{C}_2$, de même diamètre D. Ces deux solides sont placés en tandem et sont séparés par une distance L. Le domaine d'étude est de taille $40D \times 20D$. Les deux solides sont centrés verticalement sur le domaine, et le cylindre $\mathcal{C}_1$ est situé à une distance de $10D$ du bord amont (voir Figure 3.19). Le pas d'espace est choisi tel que $\triangle x = \triangle y = 1.6 * 10^{-2}$ et le pas de temps vaut $\triangle t = 10^{-3}$, respectant ainsi la condition CFL.

Nous allons évaluer les coefficients de trainée et nombre de Stroual, noté St, pour différents nombres de Reynolds et pour différentes valeurs de L. Nous comparerons ensuite nos résultats à ceux de la littérature.

Les résultats obtenus sont rassemblés dans les tables 3.2 ,3.3 et 3.4, et montrent une bonne concordance avec les résultats de la littérature.

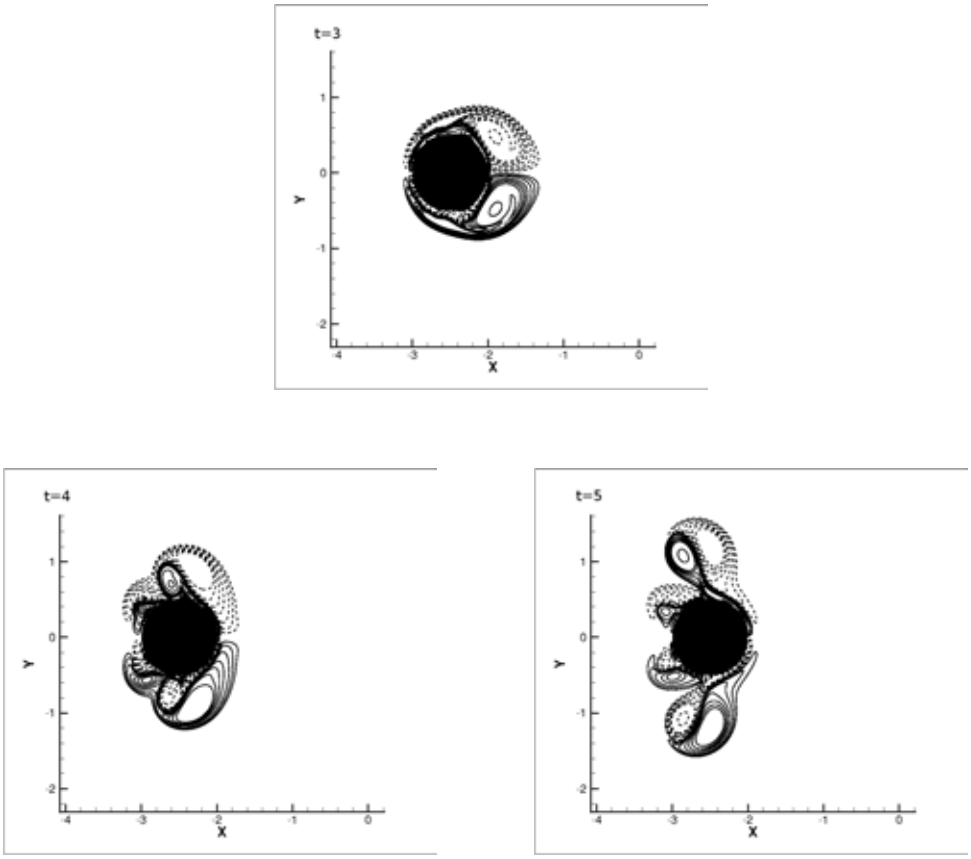


FIGURE 3.17 – Isolignes de vorticité pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$.

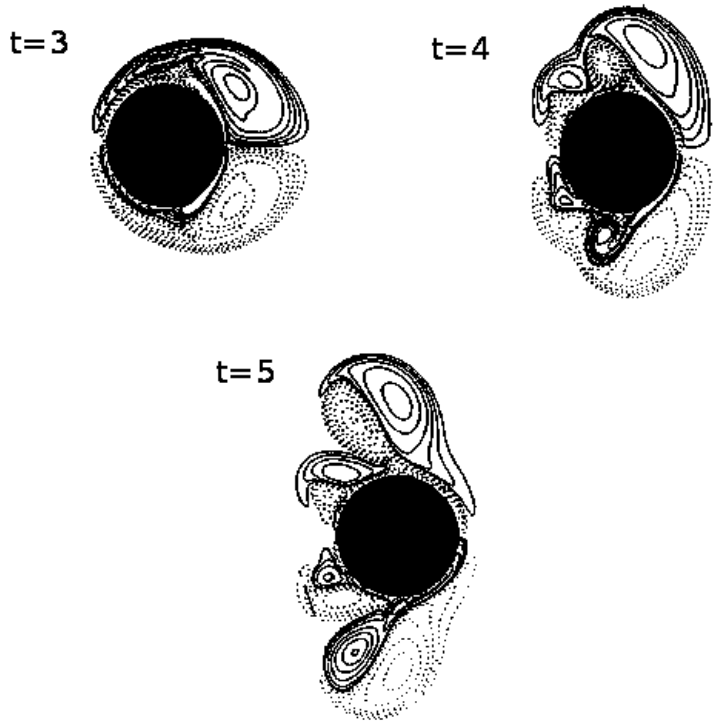


FIGURE 3.18 – Isolignes de vorticité pour un cylindre mis en mouvement instantanément arrêté au temps $t = 2.5$ à $Re = 550$ présenté par Koumoutsakos *et al.* [61].

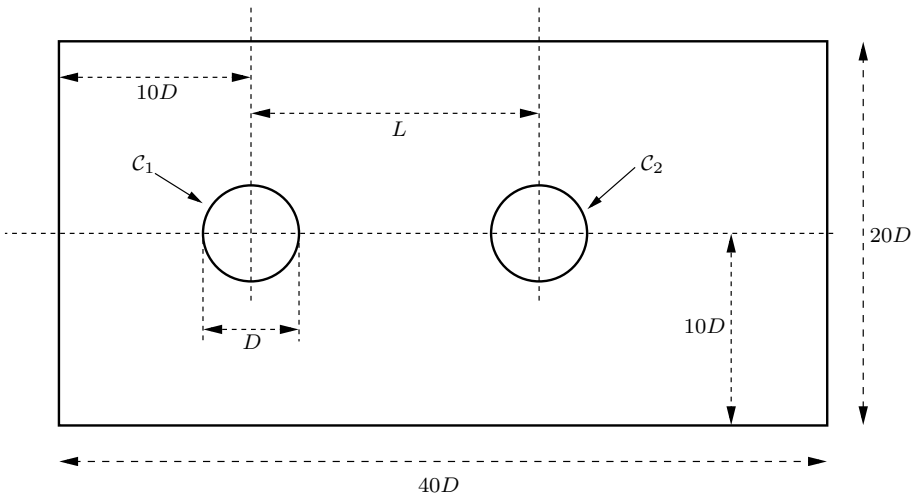


FIGURE 3.19 – Représentation des différents éléments géométriques pour la modélisation des cylindres en tandem.

$Re = 100$								
	$L = 2.5D$				$L = 5.5D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Présente étude	1.16	−0.057	0.12	0.12	1.31	0.90	0.167	0.167
Dehkordi *et al.* [32]	1.24	−0.071	–	–	1.33	1.00	0.165	0.165
Mittal *et al.* [73]	1.27	−0.075	–	–	1.43	0.86	0.168	0.168
Ding *et al.* [34]	1.16	−0.09	–	–	1.33	0.95	0.160	0.160
Sharman *et al.* [95]	1.17	−0.062	0.118	–	1.25	0.65	0.154	0.154

TABLE 3.2 – Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 100$ et $L = 2.5D$ ou $L = 5.5D$.

$Re = 200$								
	$L = 1.5D$				$L = 2D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Présente étude	1.04	−0.17	0.185	0.185	1.00	−0.19	0.140	0.140
Dehkordi *et al.* [32]	1.05	−0.15	0.175	0.175	1.03	−0.16	0.138	0.138
Meneghini *et al.* [68]	1.06	−0.18	0.167	0.167	1.03	−0.17	0.130	0.130
Mahir *et al.* [67]	–	–	–	–	1.06	−0.21	–	–
Slaouti *et al.* [96]	–	–	–	–	0.89	−0.21	0.130	0.130

TABLE 3.3 – Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 200$ et $L = 1.5D$ ou $L = 2D$.

$Re = 200$								
	$L = 3D$				$L = 4D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Présente étude	0.96	−0.18	0.126	0.126	1.23	0.89	0.185	0.185
Dehkordi *et al.* [32]	1.00	−0.08	0.129	0.129	1.16	0.52	0.18	0.18
Meneghini *et al.* [68]	1.00	−0.08	0.125	0.125	1.18	0.38	0.17	0.17
Mahir *et al.* [67]	1.05	−0.56	0.130	0.130	1.34	0.56	0.18	0.18
Slaouti *et al.* [96]	0.87	−0.16	–	–	1.11	0.88	0.19	0.19

TABLE 3.4 – Comparaison du coefficient de trainée et du nombre de Strouhal pour deux cylindres en tandem avec des études précédentes pour $Re = 200$ et $L = 3D$ ou $L = 4D$.

Nous constatons que nos résultats sont concordants avec la littérature dans le cas d'un cylindre fixe ou dans le cas où nous imposons son déplacement. La question se pose maintenant de connaître le comportement du solveur lorsque le déplacement du solide est déterminé par des forces extérieures. Par conséquent, nous avons décidé de modéliser la sédimentation d'une bille dans une cavité.

3.3.4 Sédimentation d'une bille

On étudie donc le cas de la chute d'une cylindre circulaire 2D sous l'effet de la gravité dans une cavité. On se place dans la même configuration que celle présentée par Coquerelle et Cottet dans [30]. On considère donc un domaine d'étude de taille $[0; 2] \times [0; 6]$ dans lequel se situe un cylindre centré initialement en $(1; 4)$, de rayon $r = 0.25$ et de masse volumique $\rho_s = 1.5$. Ce cylindre est entouré d'un fluide de viscosité cinématique $\nu = 0.01$ et de masse volumique $\rho_f = 1$. Le pas d'espace utilisé pour les simulations est $\triangle x = \triangle y = 1/256$ et le pas de temps est $\triangle t = 10^{-5}$.

À travers les figures 3.20, 3.21 et 3.22, nous pouvons constater que nous obtenons une vitesse verticale en norme supérieure à celle de Coquerelle et Cottet. Les profils sont très similaires jusqu'au temps $t = 0.1$, puis la vitesse verticale que nous obtenons devient plus importante que la leur, la nôtre étant de 13.4 lorsque celle de Coquerelle et Cottet est légèrement supérieure à 12. Par conséquent, notre bille atteint le fond de la cavité plus tôt. Notons que sur la figure 3.20, nous avons deux points au bord du cylindre ayant une vitesse très inférieure aux autres points du cylindre. Il s'agit des deux ghost points soumis à l'IPC.

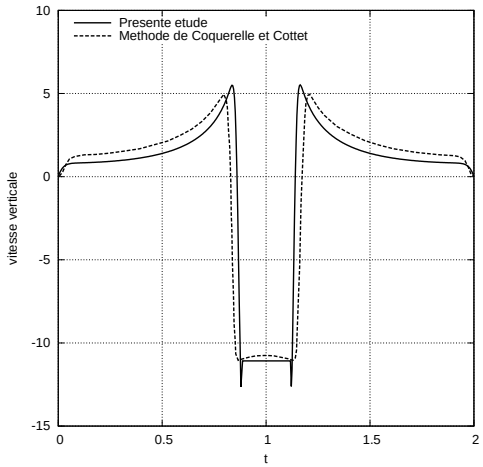


FIGURE 3.20 – Comparaison des coupes de vitesses verticales au niveau du cylindre au temps $t = 0.1$ entre notre étude et celle de Coquerelle et Cottet [30].

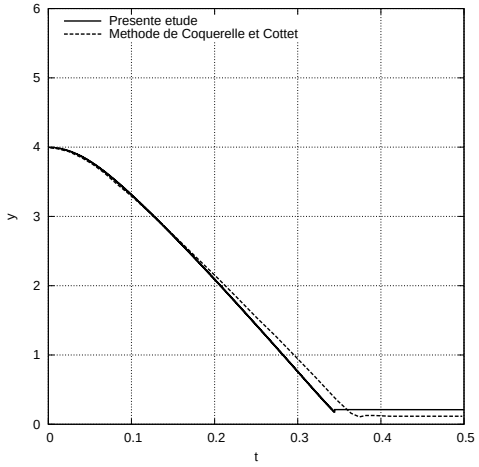


FIGURE 3.21 – Comparaison entre notre étude et celle de Coquerelle et Cottet [30] des évolutions en temps de la position du centre de la bille.

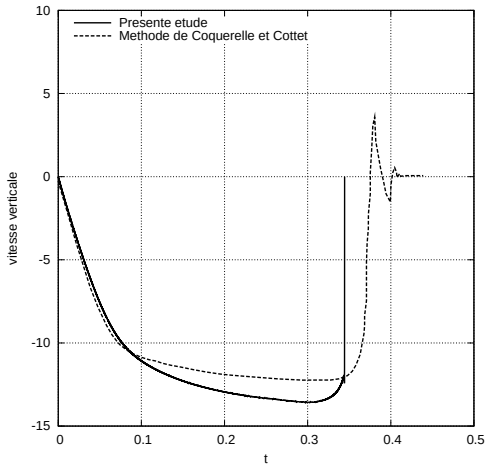


FIGURE 3.22 – Comparaison entre notre étude et celle de Coquerelle et Cottet [30] des évolutions en temps de la vitesse verticale du centre de la bille.

3.4 Conclusion

À travers ce chapitre, nous avons présenté une nouvelle méthode de type frontières immergées basée sur une fusion entre les approches Ghost-Cell et pénalisation.

Nous avons choisi la méthode de pénalisation pour sa simplicité d'implémentation et la méthode Ghost-Cell pour sa précision. De plus, ayant choisi un schéma numérique à pas fractionnaire de type prédicteur-correcteur pour la résolution des équations de Navier Stokes, la méthode de pénalisation apporte certains avantages : tout d'abord, elle nous permet d'étudier des corps solides en mouvement sans traitement particulier sur les points nouvellement fluides. De surcroît, elle permet une application des conditions de bord de pression sur l'interface fluide-structure de manière totalement implicite.

Nous nous sommes également inspirés de la méthode Ghost-Cell pour introduire notre nouvelle méthode : la *méthode IPC*. Celle-ci consiste à corriger la valeur de pénalisation appliquée aux ghost-points afin d'obtenir la bonne vitesse à l'interface en utilisant la valeur du point image. Cette correction s'effectue en cherchant une conservation du gradient de la vitesse de part et d'autre de l'interface, dans la direction normale à l'interface. Cette méthode, simple dans son concept, impose toutefois une valeur physiquement erronée sur les ghost points, et, en raison de notre choix d'une méthode de résolution des équations de Navier Stokes de type splitting, notre approche nécessite une étape de pénalisation classique supplémentaire.

La correction appliquée à la valeur de pénalisation des ghost points a pour objectif une montée en ordre en espace. Nous avons donc étudié l'ordre de convergence de notre méthode et avons pu constater que nous aboutissions sur une méthode du second ordre, aussi bien globalement que localement, en vitesse.

Une fois cet ordre de convergence établi, nous avons cherché à valider les solutions numériques obtenues par notre solveur en les comparant à des résultats issus de la littérature pour des cas tests canoniques.

Nous avons donc tout d'abord considéré le cas d'un cylindre rigide, fixe, soumis à un écoulement. Nous avons alors évalué les coefficients aérodynamiques de trainée et de portance pour des écoulements à nombres de Reynolds différents. Nous avons alors constaté une bonne concordance entre nos résultats et ceux des précédentes études.

Dans un second temps, nous nous sommes positionnés en un point de vue opposé : nous considérons un fluide au repos et un cylindre rigide mis en mouvement en son sein. Nous avons alors observé des comportements similaires pour les deux points de vue et avons également pu observer que les oscillations sur l'évolution du coefficient de trainée dues au déplacement du cylindre étaient moins importantes dans le cas de la méthode IPC que dans le cas de la pénalisation classique.

Nous avons ensuite considéré le cas d'un cylindre en mouvement qui est instantanément arrêté. Les résultats fournis par le solveur furent en très bon accord avec ceux de Koumoutsakos *et al.* [61], et ce, pour divers nombres de Reynolds.

L'étape suivante fut la validation du code pour le cas de plusieurs corps solides. Nous avons donc étudié la configuration de deux cylindres de même diamètre positionnés en tandem, dont nous avons fait varier l'écartement ainsi que le nombre de Reynolds associé à l'écoulement. Une nouvelle fois, nos résultats ont concordé avec la littérature.

La dernière configuration étudiée fut la chute d'une bille dans une cavité sous l'effet de la gravité. Cette dernière étude fut celle qui apporta un bémol. En effet, nous avons obtenu un maximum de vitesse verticale est supérieur de 10% à celui de Coquerelle et Cottet [30]. Cet écart pout être expliqué par l'ordre des schémas utilisés. En effet, Coquerelle et Cottet ont utilisé une méthode vortex d'ordre 1 tandis que la méthode IPC est du second ordre. Le premier ordre surestime les effets visqueux, ce qui peut expliquer que notre bille ait une vitesse verticale maximale supérieure à la leur.

Ayant validé la méthode IPC pour des corps rigides, nous allons maintenant l'appliquer à la simulation de corps déformables.

Nous avons alors choisi de modéliser des nageurs, en l'occurrence des poissons. C'est cette étude qui constitue la seconde partie de ce manuscrit.

Deuxième partie

Application à la Simulation de la Nage

Chapitre 4

Simulation Bidimensionnelle de la Nage

Sommaire

4.1 Définition du poisson 86
4.1.1 Les différents modes de nage 86
4.1.2 Les différentes catégories de poisson à nage ondulatoire 88
4.1.3 Définition du modèle géométrique du poisson 91
4.2 Optimisation 95
4.2.1 Éléments nécessaires à une bonne formulation d'un problème d'optimisation 95
4.2.2 Les méthodes de type déterministe 96
4.2.3 Les méthodes non déterministes 100
4.3 Optimisation de la nage 103
4.3.1 Optimisation des paramètres de nage 104
4.3.2 Trois paramètres de contrôle 113
4.3.3 Optimisation de la forme et de la nage du poisson 115
4.4 Conclusion 115

À l'heure actuelle, l'homme n'est pas parvenu à atteindre les performances sous-marines des poissons. Il essaie d'égaler les capacités des poissons sur trois niveaux :

- l'accélération,
- la vitesse,
- la manoeuvrabilité.

En effet, certains poissons comme le requin mako (ou requin taupe-bleu) sont connus pour leur surprenante capacité d'accélération. Ils sont également très rapides, pouvant atteindre les 90 km/h. Ils peuvent également faire des virages de 180° sans perte de vitesse et avec des rayons de courbure variant entre 10 et 30% de la longueur totale de leur corps. En comparaison, pour effectuer les virages, des bâteaux doivent réduire de près de 50% leur vitesse et les rayons de courbure sont de l'ordre de plusieurs longueurs du bâteau.

Ce sont toutes ces remarquables performances qui ont poussé les ingénieurs à s'engager dans le biomimétisme. En l'occurence, à concevoir des robots sous-marins reproduisant la forme et la nage des poissons.

Notons ici un autre facteur important dans l'efficacité de la nage des poissons que nous ne prendrons pas en compte dans notre modélisation. Il s'agit des écailles présentes sur le corps de certains poissons. En effet, les écailles du poisson lui permettent de réduire la trainée s'appliquant sur son corps en séparant les flux d'eau. De la même manière qu'une

balle de golf possède une surface alvéolée pour un meilleur aérodynamisme, les écailles de poissons rendent la nage plus efficace. Ainsi, sur le requin mako par exemple, on notera la présence d'écailles sur certaines parties de son corps.

Ce facteur ne sera pas pris en compte dans notre étude mais constitue une possible perspective à prendre en considération dans une future étude.

Nous allons à présent présenter les différents types de poissons associés à la proportion de leur corps oscillant durant la nage. Nous verrons alors quel type de poisson et quel type de nage nous avons choisi de modéliser ainsi que la démarche mathématique que nous avons choisie de suivre.

L'un de nos buts étant l'optimisation de la forme et de la nage du poisson suivant un certain objectif que nous définirons par la suite, nous présenterons une vue d'ensemble des méthodes d'optimisation. Enfin, nous verrons quelles méthodes nous avons abordées et quels ont été les résultats obtenus.

4.1 Définition du poisson

Avant de décrire les différentes catégories de poisson, définissons les différents modes de nage que ceux-ci peuvent adopter. En effet, comprendre dans un premier temps les nages existantes aidera à distinguer les différentes classes de poisson.

4.1.1 Les différents modes de nage

Nous nous baserons sur la classification de Breder [20] [1] qui distingue deux types de nage :

- la nage périodique,
- la nage transitoire.

La nage périodique se caractérise par la répétition cyclique d'un mouvement de type propulsion. Elle est utilisée par les poissons couvrant de longues distances à vitesse approximativement constante.

La nage transitoire comporte les départs rapides, les virages et les plongées. Elle correspond aux comportements de chasse et de fuite.

Cette dernière catégorie est plus rarement modélisée que la nage périodique étant donné qu'il s'agit de mouvements difficiles à valider numériquement puisqu'ils sont très rarement effectués plusieurs fois de la même manière.

Dans notre étude, nous concentrerons sur la nage de type périodique.

Au sein de la nage périodique, on distingue ensuite deux modes de propulsion.

Le premier mode de propulsion est appelé BCF ("Body and/or Caudal Fin"). Il s'applique aux poissons courbant leur corps et/ou leur nageoire caudale (voir Figure 4.1) créant un va-et-vient qui les propulse.

1. Il existe d'autres types de classification, plus riches généralement en catégories. Voir par exemple [116].

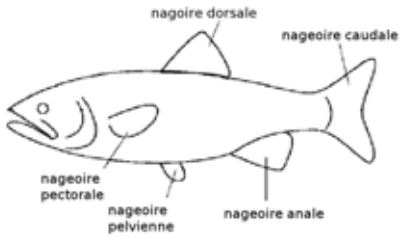


FIGURE 4.1 – Terminologie des différentes nageoires d'un poisson [94].

D'autres poissons nagent grâce aux mouvements de leurs nageoires dorsales, anales (les deux formant les nageoires dites "médianes"), pectorales et pelviennes (les deux formant les nageoires dites "paires") (voir Figure 4.1). On appelle cette nage le mode MPF ("Median and/or Paired Fin"). Approximativement 15% des poissons utilisent le mode MPF. On se concentrera donc sur le mode BCF dans notre modélisation.

À l'intérieur de ces modes, existent deux mouvements caractéristiques :
- le mouvement ondulatoire,
- la mouvement oscillant.

On caractérise le mouvement ondulatoire par la propagation d'une onde rétrograde à la direction du mouvement, d'amplitude croissante le long du corps du poisson. Ce mouvement existe dans le mode BCF (exemple : l'anguille) et le mode MPF (exemple : la raie pastenague) (voir Figure 4.2 [2]).

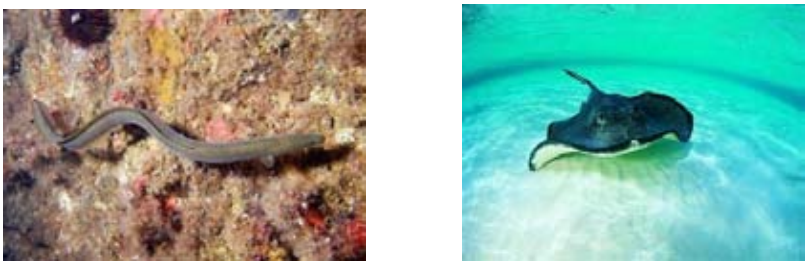


FIGURE 4.2 – L'anguille à gauche et la raie pastenague à droite.

Le mouvement oscillant se caractérise simplement par un battement stationnaire, aucune onde n'intervient dans ce mouvement. Cette catégorie de poisson est appelée osctraciiforme. Le baliste (voir Figure 4.3 [3]) est un exemple de poisson utilisant ce type de mouvement.

2. sources : aappma-molsheim.com/pages/infos/qu-est-ce-qu-une-anguille-jaune-et-une-anguille-argentee.html
www.ocearium-croisic.fr/pages/3-oceans/le-tunnel/la-raie-pastenague.php
3. source : www.aquaportail.com/definition-4703-baliste.html



FIGURE 4.3 – Le baliste.

Dans notre cas, nous nous concentrerons sur les mouvements de type ondulatoire.

Les différentes catégories de poissons utilisant le mode de propulsion BCF à mouvement ondulatoire sont les suivantes :

- anguilliforme,
- subcarangiforme,
- carangiforme,
- thunniforme.

Elles se différencient par la proportion du corps du poisson déformé lors de la nage (voir Figure 4.4).

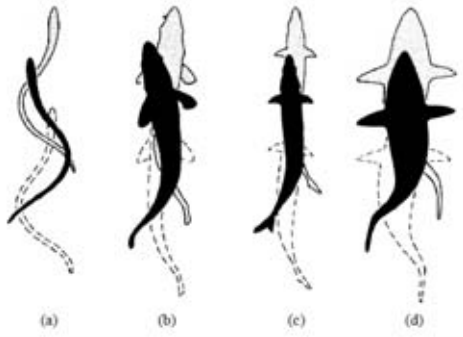


FIGURE 4.4 – Evolution de la proportion du corps déformé lors de la nage de type BCF : (a) poisson anguilliforme, (b) poisson subcarangiforme, (c) poisson carangiforme et (d) poisson thunniforme [94].

4.1.2 Les différentes catégories de poisson à nage ondulatoire

Les différentes catégories de poisson sont résumées dans la figure 4.5. Nous allons détailler brièvement ci-après les poissons à nage ondulatoire.

Les poissons de type anguilliforme

Chez les poissons de type anguilliforme tels que l'anguille, le congre ou la murène (voir

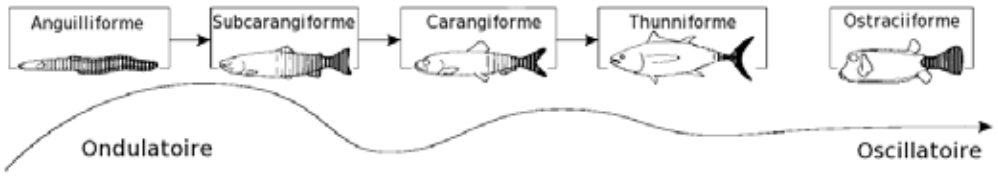


FIGURE 4.5 – Les différentes catégories de poisson [94].

Figure 4.6 [4]), la totalité de leur corps est traversée par l'onde. Cela leur procure une très grande capacité d'accélération. De plus, ils ont aussi la capacité de pouvoir également se déplacer en marche arrière en inversant simplement la direction de propagation de l'onde. Ce sont toutefois des poissons qui ne peuvent pas atteindre de grandes vitesses de pointe : environ 10km/h.



FIGURE 4.6 – L'anguille à gauche et la murène à droite.

Les poissons de type subcarangiforme

Chez les poissons de type subcarangiforme, l'amplitude de l'oscillation est plus faible et ne s'étend pas sur tout le corps, s'arrêtant généralement à la moitié postérieure de la longueur du poisson, le reste du corps étant rigide. Les truites et saumons (voir Figure 4.7 [5]) s'inscrivent dans cette famille de poissons. Leur vitesse est beaucoup plus variable suivant les espèces. Par exemple, la vitesse de pointe des truites avoisine les 35 km/h tandis que celle du saumon se situe autour des 45 km/h.

Les poissons de type carangiforme

La rigidité des poissons de type carangiforme est encore plus prononcée : seul le premier tiers de leur corps peut se tordre au maximum. Ils sont néanmoins plus rapides que les poissons anguilliformes ou subcarangiformes, bien que leur manoeuvrabilité et leur capacité d'accélération soient moindres, dues à cette forte rigidité.

Pompanos, requins marteaux, requins bleus et sardines constituent des exemples de

4. sources : www.mysterra.org/webmag/murenes.html
piski.free.fr/poissons/fichepoisson.php ?id=8

5. sources : www.entrepecheurs.com/encyclopedie/poissonsd-eau-douce/truite-de-mer
www/lefigaro.fr/environnement/2011/10/18/01029-20111018ARTFIG00764-des-saumons-gravement-contamines-au-canada.php



FIGURE 4.7 – La truite à gauche et le saumon à droite.

poissons carangiformes (voir Figure 4.8 [6]).

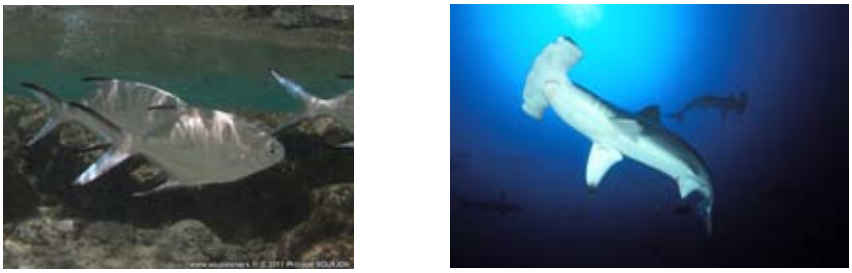


FIGURE 4.8 – Le pompano à gauche et le requin marteau à droite.

Les poissons de type thunniforme

On appelle également cette classe de poisson "carangiforme modifiée". Ce sont les poissons les plus efficaces en terme de rapidité. Le plus rapide d'entre eux est l'espadon voilier qui atteint des pointes de vitesse à près de 110 km/h.

Seule la nageoire caudale génère les ondulations chez les poissons thunniformes. Ils gèrent cependant difficilement les faibles vitesses et les virages serrés.

La classe des poissons thunniformes comprend entre autre le requin blanc, les thons et tous les types de maquereaux (voir Figure 4.9 [7]). Leur gamme de vitesse est très large car le maquereau atteint les 30km/h, le thon frôle les 75km/h, tandis que les vitesses du requin blanc et de l'espadon avoisinent les 100km/h.

Notons enfin, et cette remarque possède des exceptions, qu'en règle générale, plus la proportion ondulante du corps du poisson est grande, plus le poisson est fin (exemple d'exception : le maquereau).

Nous avons donc choisi de modéliser des poissons au mode BCF à nage périodique. Nous avons également choisi d'exclure les poissons présentant des nageoires autres que

6. sources : fran.cornu.free.fr/affichage/affichage_nom.php ?id_espece=1414
www.linternaute.com/nature-animaux/animaux/diaporama/festival/image-sous-marine/2.shtml

7. sources : francois.derouin.free.fr/svtclggsand/ ?p=4219
www.rtl.be/people/album/341/#image-8124/full :1




FIGURE 4.9 – Le thon rouge à gauche et l'espadon voilier à droite.

caudale, significatives. Nous n'étudierons donc pas des poissons comme le requin ou l'espadon, mais plutôt des poissons tels que la truite, la sardine, le saumon ou le maquereau.

4.1.3 Définition du modèle géométrique du poisson

Nous définirons tout d'abord le modèle géométrique choisi pour le poisson, puis sa loi de nage.

Définition géométrique

On considère un poisson que l'on observe de dessus. La géométrie que l'on observera sera alors telle une aile d'avion.

La paramétrisation choisie pour représenter une aile d'avion suit celle présentée par Bergmann et Iollo dans [13]. On appliquera donc une transformation de Karman-Trefftz.

La transformation de Karman-Trefftz consiste à transformer un cercle de rayon $r_c = 1$ en une aile d'avion de longueur 1 avec un angle à la queue réglable. La procédure de transformation est la suivante :

On considère le cercle $\mathcal{C}_c$ de centre ζ_c et de rayon $r_c = 1$, avec $\zeta_c = \eta_c + i\theta_c$.
Soit $\zeta \in \mathcal{C}_c$ tel que $\zeta = \eta + i\theta$.

On définit également le paramètre α, correspondant à l'angle au niveau du bord de fuite. On introduit ensuite n tel que $\alpha = (2 - n)\,\pi$.

La transformée de Karman-Trefftz consiste à définir $z = x + iy$ tel que :

$$z = n\frac{\left(1 + \frac{1}{\zeta}\right)^n + \left(1 - \frac{1}{\zeta}\right)^n}{\left(1 + \frac{1}{\zeta}\right)^n - \left(1 - \frac{1}{\zeta}\right)^n}.$$

Le résultat obtenu est une aile d'avion de longueur 1, d'angle α au bord de fuite.

Pour obtenir une aile de longueur l, il suffit d'appliquer une homothétie de facteur l au profil.

Dans notre cas, nous souhaitons avoir un profil au repos symétrique par rapport à l'axe des abscisses. On prendra donc $\theta_c = 0$ (voir Figure 4.10).

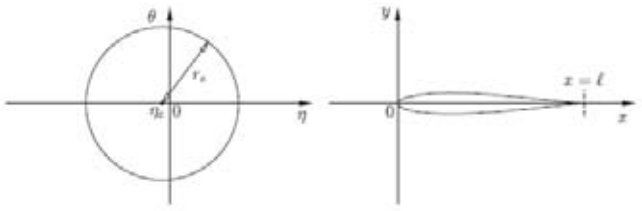


FIGURE 4.10 – Transformation de Karman-Trefftz d'un cercle en un profil d'aile de longueur l : à gauche le cercle initial, à droite le profil d'aile obtenu après transformation [13].

De plus, l'abscisse η_c du centre du cercle $\mathcal{C}_c$ influencera l'épaisseur du poisson créé (voir Figure 4.11). Par exemple,

- $\eta_c = -0.01$ formera des poissons ayant une forme de type anguilliforme,
- $\eta_c = -0.04$ formera des poissons ayant une forme de type thunniforme,
- $-0.04 < \eta_c < -0.01$ formera des poissons ayant une forme de type carangiforme à subcarangiforme.

Par conséquent, on obtient un profil d'aile pour notre représentation 2D du poisson dépendant des trois paramètres suivants :

- η_c l'épaisseur du poisson,
- α l'angle à la queue du poisson,
- l la longueur du poisson.

Définition de la loi de nage

On considère un poisson de longueur $l = 1$ dont le nez se situe en $x = 0$.
On souhaite une loi de nage pour nos poissons qui vérifie les propriétés suivantes :

- la loi de nage est périodique,
- la loi de nage est de type ondulatoire,
- l'enveloppe de la loi de nage est croissante le long du corps du poisson.

De plus, afin d'être en mesure de modéliser les quatre types de poisson, on doit être capble de ne déformer qu'une certaine proportion du corps du poisson.

Une nouvelle fois, nous avons choisi de suivre la loi de nage utilisée dans [13].

Leur idée consiste à déformer la colonne vertébrale du poisson grâce à une loi de déformation choisie, puis à reconstruire le poisson.

La loi de déformation de la colonne vertébrale du poisson s'écrit :

$$y(x,t) = a(x)\, sin\left(2\pi\left(\frac{x}{\lambda} + ft\right)\right), \tag{4.1}$$

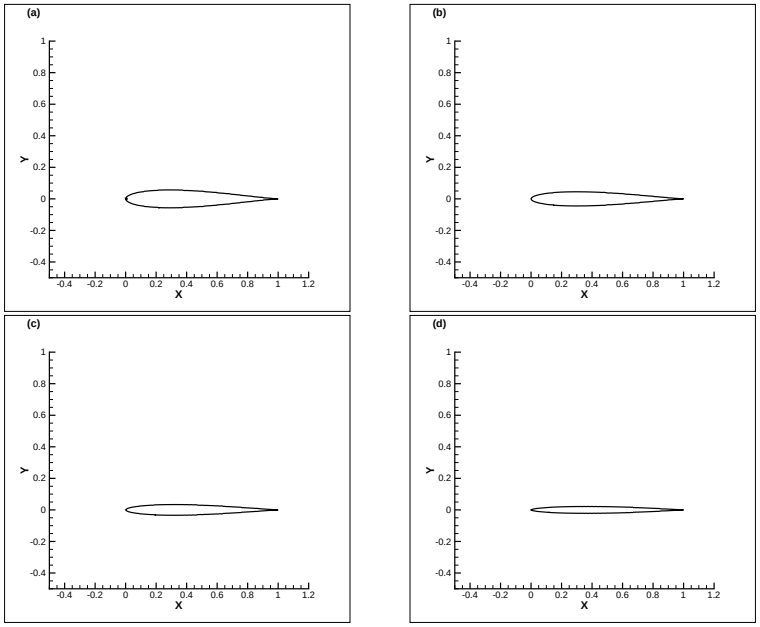


FIGURE 4.11 – Géométries de poisson obtenues pour différentes valeurs de η : (a) $\eta = -0.004$, (b) $\eta = -0.003$, (c) $\eta = -0.002$ et (d) $\eta = -0.001$.

où $\begin{cases} y \text{ correspond aux ordonnées des points de la colonne vertébrale déformée,} \\ x \text{ correspond aux abscisses des points de la colonne vertébrale déformée,} \\ \lambda \text{ est la longueur d'onde de la nage,} \\ f \text{ est la fréquence de la nage,} \\ t \text{ est le temps considéré,} \\ a\,(x) \text{ est l'amplitude de la nage.} \end{cases}$

L'amplitude $a\,(x)$ est définie par :

$$a\,(x) = A\left(c_1 x + c_2 x^2\right), \tag{4.2}$$

avec A l'amplitude maximale de la nage, c_1 et c_2 des coefficients de nage variables tels que $c_1 + c_2 = 1$. En conséquence, $a\,(x)$ définit l'enveloppe de la colonne vertébrale du poisson déformé ; elle est bien croissante le long du corps du poisson et, en ajustant les paramètres c_1 et c_2, on peut choisir la proportion du corps qui sera déformé.

Ainsi, dans [13], on peut voir que :
- si $c_1 = 0.1$ et $c_2 = 0.9$, alors le poisson aura une nage de type thunniforme,
- si $c_1 = 1.0$ et $c_2 = 0$, alors le poisson aura une nage de type anguilliforme,
- si $c_1 = 0.4$ et $c_2 = 0.6$, alors le poisson aura une nage de type carangiforme,
- si $c_1 = 0.7$ et $c_2 = 0.3$, alors le poisson aura une nage de type subcarangiforme.

Notons enfin que, bien entendu, l'abscisse curviligne $s\,(x)$ de la colonne vertébrale déformée ne doit pas excédée la longueur du poisson. Par conséquent, dans le cas général d'un poisson de longueur l dont le nez se situe en $x = x_0$, l'abscisse curviligne de la colonne

vertébrale déformée du poisson s'écrit :

$$s(x) = \int_{x_0}^{x} \sqrt{1 + \left(\frac{\partial y(x', t)}{\partial t} \right)^2} dx'. \tag{4.3}$$

D'où, déterminer x_{max} tel que $s(x_{max}) = l$ revient à inverser la fonction s et déterminer $s^{-1}(l)$.

Ainsi, Bergmann et Iollo obtinrent, entre autre, dans [13] les résultats suivants : pour un Reynolds[8] de 10^4, le poisson le plus rapide est le poisson de type thunniforme, puis le carangiforme, le subcarangiforme et l'anguilliforme. En outre, le poisson ayant la plus grande capacité d'accélération est l'anguilliforme, puis le subcarangiforme, le carangiforme et le thunniforme, tout ceci étant cohérent avec la réalité.

Par conséquent, nous obtenons une modélisation de la nage du poisson dépendante de quatre paramètres :

- c_1 (ou c_2) l'un des coefficients d'enveloppe de la nage,
- A l'amplitude de la nage,
- λ la longueur d'onde de la nage,
- f la fréquence de la nage.

Finalement nous avons donc sept paramètres : les trois paramètres de géométrie (η, α et l) et les quatre paramètres de nage (c_1, A, λ et f).

Afin d'observer les effets des diverses géométries et lois de nage, nous avons représenté en figure 4.12 la vorticité créée pour quatre nageurs différents dont les caractéristiques sont données en table 4.1.3. Dans nos paramètres, nous avons choisi une longueur de corps, une amplitude et un angle à la queue identiques pour tous les poissons. La figure 4.12 montre bien la création d'allées de Von Karman dont les amplitudes diffèrent suivant le nageur considéré.

	Forme			Nage				
Poisson	η_c	α	l	A	c_1	c_2	f	λ
thunniforme	-0.04	6	1	0.2	0	1	3	1.5
carangiforme	-0.03	6	1	0.2	0.3	0.7	2	1.25
subcarangiforme	-0.02	6	1	0.2	0.7	0.3	1.5	1
anguilliforme	-0.01	6	1	0.2	1	0	1	0.5

TABLE 4.1 – Paramètres utilisés pour la forme et la nage pour les différents poissons.

8. Sfakiotakis *et al.* expliquent dans [94] que le nombre de Reynolds pour de telles simulations est compris entre 10^3 et 5.10^6.

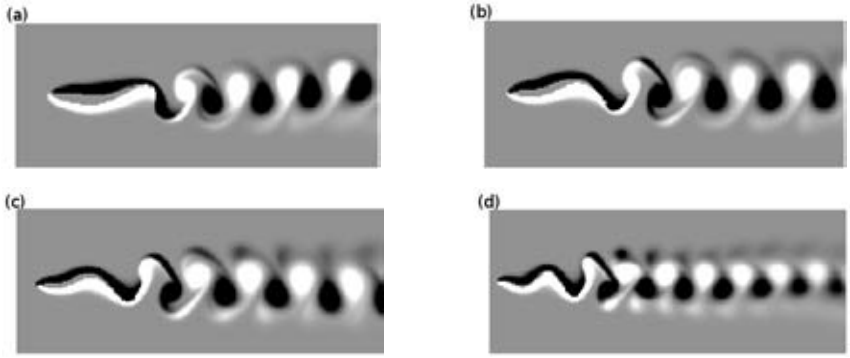


FIGURE 4.12 – Simulation de la nage des quatre types de poisson au mode de nage BCF : (a) thunniforme, (b) carangiforme, (c) subcarangiforme et (d) anguilliforme.

Notre objectif à présent est de déterminer la forme et la nage optimales dans un objectif donné (exemple : vitesse maximum, puissance dépensée minimum etc..). Par conséquent, après avoir présenté brièvement un aperçu des méthodes d'optimisation existantes, nous verrons quelle(s) méthode(s) nous avons choisie(s) d'utiliser et quels ont été les résultats obtenus.

4.2 Optimisation

L'objectif de cette section n'est pas de présenter l'ensemble des méthodes d'optimisation tant il en existe et la tâche serait grande. Nous souhaitons ici introduire les éléments essentiels à tout problème d'optimisation et présenter quelques méthodes déterministes et non déterministes, afin d'avoir un aperçu des approches existantes et de leur grande variété. Cela nous permettra de mieux comprendre quelle(s) méthode(s) nous avons choisie(s) par la suite de tester.

En conséquence, nous verrons dans un premier temps quels sont les éléments indispensables à tout problème d'optimisation. Puis nous proposerons un aperçu des deux grandes catégories de méthode de résolution : les méthodes basées sur le gradient d'une fonction objectif, appelées méthodes déterministes, et les méthodes stochastiques dépendantes d'un caractère aléatoire.

4.2.1 Éléments nécessaires à une bonne formulation d'un problème d'optimisation

Afin qu'un problème d'optimisation soit mathématiquement bien posé, il doit présenter certains éléments indispensables :

- **Les variables d'état ϕ :**
  Il s'agit des grandeurs physiques dont le problème est dépendant (exemple : pression, vitesse, température...).
- **Les paramètres de contrôle c :**
  Les paramètres de contrôle peuvent prendre diverses formes suivant le problème d'optimisation considéré. Il peut s'agir de paramètres décrivant la forme à optimiser, de conditions aux limites ou initiales par exemple.
- **Une fonction objectif J :**
  Il s'agit du (ou des) objectif(s) que l'on souhaite atteindre (exemple : minimisation de trainée, maximisation de vitesse, minimisation de l'énergie...). La fonction objectif dépend explicitement des variables d'état et des paramètres de contrôle. Son choix est essentiel pour que le problème soit bien posé.
- **Les contraintes physiques $\mathcal{F}$:**
  Celles-ci traduisent l'évolution des variables d'état en fonction des paramètres de contrôle (par exemple : les équations de Navier Stokes avec conditions de bord et initiales) et forment les équations dites d'état ou de contrainte.

Une fois tous ces éléments introduits, on peut définir le problème d'optimisation de la manière suivante :

Déterminer les variables d'état et paramètres de contrôle tels que la fonction objectif J soit minimale (ou maximale suivant l'objectif considéré), en respectant les contraintes $\mathcal{F}$.

Nous avons ainsi défini un problème d'optimisation sous contraintes.
Les approches pour résoudre ce problème peuvent être de deux types :
- déterministe,
- stochastique.

4.2.2 Les méthodes de type déterministe

Ces méthodes sont basées sur une évaluation du gradient de la fonction objectif par rapport aux paramètres de contrôle. Cette tâche peut s'avérer particulièrement compliquée et constituer un obstacle dans certains cas.

Après avoir déterminer le gradient de la fonction objectif, la résolution s'effectue de manière itérative en suivant la direction de descente donnée par le gradient.

L'inconvénient de ces méthodes est qu'elles ne convergent *a priori* que vers un minimum (ou un maximum) local, et ce, au prix d'un nombre important d'itérations. Autrement dit, un grand nombre de résolutions des équations d'état doit être effectué, ce qui peut coûter cher en temps d'exécution, ainsi qu'en place mémoire.

Nous allons à présent présenter quelques méthodes de détermination du gradient de la fonction objectif, puis nous verrons la démarche générale d'un algorithme de descente.

a) Détermination du gradient de la fonction objectif

Nous allons présenter deux méthodes de détermination du gradient de la fonctionnelle J :

- la méthode des sensibilités,
- la méthode adjointe.

(i) La méthode des sensibilités

La méthode des sensibilités consiste à déterminer le gradient de la fonctionnelle J grâce à l'utilisation de petites perturbations sur les variables d'état $\{\phi_j\}$ et les paramètres de contrôle $\{c_i\}$.

En effet, on rappelle qu'on suppose que J dépend explicitement des variables $\{\phi_j\}$ et des paramètres $\{c_i\}$. Pour simplifier, on suppose tout d'abord $i = j = 1$.

Par conséquent, déterminer ∇J revient à calculer $\dfrac{dJ}{dc}(\phi, c)$,
soit,

$$\frac{dJ}{dc} = \frac{\partial J}{\partial \phi}\frac{d\phi}{dc} + \frac{\partial J}{\partial c}. \tag{4.4}$$

Ayant l'expression analytique de J, on sait donc calculer $\dfrac{\partial J}{\partial \phi}$ et $\dfrac{\partial J}{\partial c}$. La difficulté se pose pour l'expression de $\dfrac{d\phi}{dc}$.

En effet, cette expression n'est pas connue *a priori* explicitement. L'idée derrière la méthode des sensibilités est d'utiliser les équations de contrainte $\mathcal{F}(\phi, c) = 0$. À partir de là, deux possibilités s'offrent à l'utilisateur :

- la détermination exacte du gradient,
- l'approximation numérique du gradient.

La détermination exacte du gradient

Grâce à l'application de petites perturbations sur ϕ et c, qui deviennent respectivement $\phi + \delta\phi$ et $c + \delta c$, en différenciant $\mathcal{F}$ et en la linéarisant (grâce à l'hypothèse de petites perturbations), on aboutit à une formulation linéaire de $\dfrac{d\phi}{dc}$.

Toute cette démarche revient donc à écrire :

sachant que $\mathcal{F}(\phi, c) = 0$, grâce au théorème des fonctions implicites, nous pouvons écrire :

$$d\mathcal{F} = \frac{\partial \mathcal{F}}{\partial \phi} \cdot \delta\phi + \frac{\partial \mathcal{F}}{\partial c} \cdot \delta c = 0.$$

D'où,

$$\frac{d\phi}{dc} = -\left(\frac{\partial \mathcal{F}}{\partial \phi}\right)^{-1} \cdot \frac{\partial \mathcal{F}}{\partial c}. \tag{4.5}$$

On aboutit alors à une formulation exacte de $\dfrac{d\phi}{dc}$, d'où de $\dfrac{dJ}{dc}$.

Notons que cette équation se transforme en un système linéaire dans le cas où $j > 1$.

La détermination approchée du gradient

Celle-ci consiste simplement à utiliser des différences finies pour déterminer $\frac{d\phi}{dc}$.
On écrit alors :

$$\frac{d\phi}{dc}(\phi, c) \simeq \frac{\phi(c + \delta c) - \phi(c)}{\delta c},$$

où $c + \delta c$ correspond à une petite perturbation du paramètre de contrôle c tel que $\mathcal{F}(\phi(c + \delta c), c + \delta c) = 0$.

L'avantage de la résolution du système exact est que, quelque soit le nombre de paramètres, une seule résolution des équations d'état permet d'obtenir le gradient de la fonctionnelle J. Toutefois, elle demande également de résoudre autant de systèmes linéaires qu'il y a de paramètres de contrôle, ce qui peut devenir couteux en pratique. La résolution exacte reste néanmoins pus performante que la résolution par différences finies qui est intrinsèquement approchée et qui est couteuse numériquement puisqu'elle demande autant de résolutions de l'équation d'état qu'il y a de paramètres de contrôle.

(ii) La méthode de l'équation adjointe

Cette approche, plus complexe, consiste à combiner l'équation adjointe du problème considéré à l'équation (4.4) pour obtenir $\frac{dJ}{dc}$.

Détermination de l'équation adjointe.

La détermination de l'équation adjointe commence en considérant les équations d'état $\mathcal{F}(\phi, c) = 0$ comme une contrainte au problème considéré. On introduit alors une fonction de Lagrange $\mathcal{L}$ définie par :

$$\mathcal{L}(\phi, c, \xi) = J(\phi, c) - \langle \mathcal{F}(\phi, c), \xi \rangle, \tag{4.6}$$

où ξ représente la variable adjointe de notre problème et est de même dimension que le nombre de contraintes associées aux équations d'état.

Notons que, bien entendu, dès que ϕ vérifie les équations d'état, on a $\mathcal{L}(\phi, c, \xi) = J(\phi, c), \forall \xi$.

Par conséquent, J est extrémale si $\mathcal{L}$ l'est également, autrement dit, si $\delta\mathcal{L} = 0$.
Soit,

$$\frac{\partial \mathcal{L}}{\partial \phi}\delta\phi + \frac{\partial \mathcal{L}}{\partial c}\delta c + \frac{\partial \mathcal{L}}{\partial \xi}\delta\xi = 0. \tag{4.7}$$

De plus, si on suppose les variables ϕ, c et ξ indépendantes alors les dérivées de Fréchet de $\mathcal{L}$ dans chacune des directions ϕ, c et ξ doivent être nulles quelles que soient

les variations $\delta\phi$, δc et $\delta\xi$. Autrement dit,

$$\frac{\partial \mathcal{L}}{\partial \phi}\delta\phi = \frac{\partial \mathcal{L}}{\partial c}\delta c = \frac{\partial \mathcal{L}}{\partial \xi}\delta\xi = 0. \tag{4.8}$$

De cette dernière égalité, nous obtenons les résultats suivants :

- La dérivée de Fréchet de $\mathcal{L}$ suivant la variable adjointe ξ nous permet de retrouver les *équations d'état* :
$$\mathcal{F}(\phi, c) = 0. \tag{4.9}$$
- La dérivée de Fréchet de $\mathcal{L}$ suivant les variables d'état ϕ nous permet d'établir l'*équation adjointe* :
$$\left(\frac{\partial \mathcal{F}}{\partial \phi}\right)^{\star} \cdot \xi = \left(\frac{\partial J}{\partial \phi}\right)^{\star}. \tag{4.10}$$
- La dérivée de Fréchet de $\mathcal{L}$ suivant les paramètres de contrôle c nous permet d'établir les *conditions d'optimalité* :
$$\left(\frac{\partial \mathcal{F}}{\partial c}\right)^{\star} \cdot \xi = \left(\frac{\partial J}{\partial c}\right)^{\star}. \tag{4.11}$$

Le système alors composé des conditions (4.9), (4.10) et (4.11) forme le système appelé *système optimal*.

Détermination du gradient de la fonction objectif J.

À partir de l'équation (4.10), on peut écrire

$$\frac{\partial J}{\partial \phi} = \xi^{\star} \cdot \frac{\partial \mathcal{F}}{\partial \phi}.$$

D'où, grâce à l'équation (4.4), on a :

$$\nabla J = \frac{dJ}{dc} = \xi^{\star} \cdot \frac{\partial \mathcal{F}}{\partial \phi}\frac{d\phi}{dc} + \frac{\partial J}{\partial c},$$

ce qui peut être réécrit en utilisant (4.5) comme [9] :

$$\nabla J = -\xi^{\star} \cdot \frac{\partial \mathcal{F}}{\partial c} + \frac{\partial J}{\partial c}. \tag{4.12}$$

En conclusion, le calcul de la fonctionnelle J s'articule autour des étapes suivantes :

1. Définition de la Lagrangienne grâce à l'équation (4.6),
2. Résolution de l'équation adjointe (4.10) fournissant les variables adjointes ξ,
3. Détermination de ∇J grâce à l'équation (4.12).

Par conséquent, on obtient ainsi le gradient de la fonctionnelle J en résolvant une seule équation, l'équation adjointe. Cela signifie que, quels que soient le nombre de paramètres $\{c_i\}$ et le nombre de variables d'état $\{\phi_j\}$, il n'y aura toujours qu'un seul système d'équations à résoudre.

9. Notons ici que la condition d'optimalité (4.11) est vérifiée si et seulement si $\nabla J = 0$.

(b) Les algorithmes à direction de descente

Connaissant à présent le gradient de la fonction objectif J par rapport aux paramètres de contrôle, on peut utiliser ce dernier pour optimiser les paramètres c, grâce à des algorithmes à direction de descente.

Supposons que l'on souhaite déterminer le minimum de la fonctionnelle J. Introduisons la notion de direction de descente.

Définition 4.1 : Une direction de descente d

On appellera d direction de descente de J dans $\mathbb{R}^n$, $n \geqslant 1$, (n correspondant au nombre de paramètres de contrôle) si et seulement si d vérifie

$$\langle \nabla J, \mathbf{d} \rangle < 0.$$

Dès lors, si d est une direction de descente de J alors, $\forall \alpha > 0$, si α est suffisamment petit, on a

$$J(c + \alpha d) < J(c). \tag{4.13}$$

α est appelé le pas de descente.

On appelle alors algorithme à direction de descente la construction d'une suite de points itérés $\{c_k\}_{k \geqslant 1}$ tels que

$$c_{k+1} = c_k + \alpha_k d_k. \tag{4.14}$$

Les différentes manières de déterminer α et d ont donné naissance à un grand nombre de méthodes.

Nous n'allons pas les présenter ici car ces méthodes ne constituent pas la démarche adoptée par la suite. Le lecteur intéressé peut néanmoins se référer à [12] pour avoir une description des approches possibles.

Toutes les méthodes à direction de descente possède néanmoins un inconvénient majeur : un fois dans un minimum local, ces méthodes ne permettent pas d'en sortir pour converger vers le minimum global.

Un solution consiste en l'utilisation de méthodes dites stochastiques. Ces méthodes permettent de sortir d'un minimum local dans lequel seraient piégées les méthodes basées sur le gradient de la fonction objectif.

4.2.3 Les méthodes non déterministes

Parmi les algorithmes non déterministes, on trouve la famille des algorithmes stochastiques.

On dit qu'un algorithme est stochastique si sa solution dépend de ses valeurs d'entrée et, dans certains cas, d'un valeur aléatoire. Généralement, de tels algorithmes utilisent les valeurs de la fonctionnelle sur un ensemble de valeurs d'entrée choisies aléatoirement, puis transforment cet ensemble à l'aide de procédures (variant suivant la méthode stochastique choisie). Le premier avantage de ces méthodes est qu'elles n'imposent pas de condition *a priori* sur la fonction objectif J. Toutefois, le premier inconvénient apparaît également :

ces méthodes ne fourniront pas nécessairement le même résultat pour des valeurs initiales différentes. Néanmoins, grâce à l'intervention d'une variable aléatoire, elles auront une probabilité plus grande de trouver le minimum global d'une fonctionnelle, que les méthodes déterministes.

Nous allons présenter deux méthodes non déterministes :

- la méthode simplexe, qui n'utilise pas de variable aléatoire,
- la méthode du recuit-simulé qui est de type stochastique.

(a) La méthode simplexe

La méthode du simplexe que nous allons présenter ici est celle introduite par Nedler et Mead [77].

Supposons que la fonction objectif J dépende de n paramètres de contrôle $\{c_i, i = 1..n\}$.

L'algorithme du simplexe se base sur le calcul initial de $n+1$ valeurs de J choisies aléatoirement. Le simplexe initial ainsi former se déplacera par la suite dans l'espace des paramètres de contrôle afin de se positionner sur le jeu de paramètres minimisant la fonctionnelle. L'algorithme complet est le suivant :

Algorithme 4.2 : Le simplexe

Initialisation : $k = 0$,
construction du simplexe initial.
Tant que critère de convergence non satisfait, faire

1. **Reflexion :**
   - Déterminer $c^k_{max} = arg\ max\ J(c_k)$,
   - Déterminer le barycentre $\bar{c}_k$ des n points restants,
   - Déterminer c^k_{ref} le réfléchi de c^k_{max} par rapport à $\bar{c}_k$:

     $$c^k_{ref} = (1+\alpha)\,\bar{c}_k - \alpha c^k_{max},$$

     (Notons que pour $\alpha = 1$, c^k_{ref} est le symétrique de c^k_{max} par rapport à $\bar{c}_k$)
   - Retirer le point c^k_{max} du simplexe et le remplacer par c^k_{ref}.

2. Test :
   - Cas 1 : si $J\left(c^k_{ref}\right) \leqslant J\left(c^k_i\right), \forall i = 1$ à n,
     (i.e. si la réflexion crée un point dont la valeur de la fonctionnelle est minimale à toutes les autres valeurs du simplexe)
     alors effectuer une **expansion** :
     - Déterminer c^k_{exp} tel que

       $$c^k_{exp} = (1-\gamma)\,\bar{c}_k + \gamma c^k_{ref}, \quad \gamma \in [1;2],$$

     - si $J\left(c^k_{exp}\right) \leqslant J\left(c^k_{ref}\right)$ alors remplacer c^k_{ref} par c^k_{exp} ,
       sinon, ne rien faire.
   - Cas 2 : si $J\left(c^k_{ref}\right) \geqslant J\left(c^k_i\right), \forall i = 1$ à n,
     (i.e. si la réflexion crée un point dont la valeur de la fonctionnelle est maximale à

toutes les autres valeurs du simplexe)
alors effectuer une **contraction** :

- Déterminer c^k_{cont} tel que

$$c^k_{cont} = (1-\beta)\,\bar{c}_k + \beta c^k_{ref}, \qquad \beta \in [0.2; 1],$$

- si $J\left(c^k_{cont}\right) \leqslant J\left(c^k_{ref}\right)$ alors remplacer c^k_{ref} par c^k_{cont} ,
  sinon, faire une **réduction** : Pour tous les points du simplexe, faire une homothétie de centre $c^k_{min} = arg\ min\ J\left(c^k_i\right)$, de rapport $\dfrac{1}{2}$.

– Cas 3 : si $J\left(c^k_{ref}\right) > J\left(c^k_{max}\right)$ Faire **réduction**
– Cas 4 : si $J\left(c^k_{ref}\right) < J\left(c^k_{max}\right)$ Faire **réflexion** puis **réduction**.

$k \leftarrow k+1$
fin tant que.

Le choix des paramètres α et β est important pour la résolution du problème. Pour éviter toute fausse convergence, ces paramètres sont généralement choisis autour de 1.

Enfin, notons que l'algorithme du simplexe est usuellement performant quand le nombre de paramètres de contrôle n'est pas trop élevé (i.e. $n \leqslant 2$). De plus, il n'existe aucune preuve mathématique de la convergence d'une fonction objectif de $\mathbb{R}^n \longrightarrow \mathbb{R}$ pour $n \geqslant 2$ [62].

(b) La méthode du recuit-simulé

Cette méthode est inspirée de la sidérurgie où, pour amener un métal à sa forme stable finale, on effectue des cuissons successives (d'où le "recuit") du métal par palier de température décroissante. À chaque palier, on transforme le métal afin de le rapprocher davantage de sa forme finale.

L'algorithme du recuit simulé simule les "recuits" d'une fonction objectif en introduisant une nouvelle variable : la température. Cette température est utilisée pour définir une probabilité : la probabilité d'acceptation d'un état moins optimal (ou "dégradé") que l'état actuel. Cette probabilité sera définie telle que plus la température est grande, plus la probabilité d'accepter un état dégradé sera importante, et réciproquement.

La probabilité p d'acceptation est de la forme

$$p = exp\left(-\frac{|\delta J|}{T}\right), \tag{4.15}$$

où $\begin{cases} \delta J \text{ est la variation de la fonction objectif entre deux états,} \\ T \text{ est la température au temps courant.} \end{cases}$

La température subit quant à elle une diminution dont la loi peut être géométrique (ex : $T \leftarrow T * 0.9$) ou arithmétique (ex : $T \leftarrow T - cste$). En règle générale, la température reste constante pendant un certain nombre d'itérations pour permettre une bonne exploration de l'espace des paramètres et éviter l'élimination prématurée de certaines configurations dégradées. Notons que plus la température sera basse, plus la zone d'exploration des paramètres sera concentrée autour de la configuration courante.

l'algorithme du recuit simulé est le suivant :

Algorithme 4.3 : Le recuit-simulé

Initialisation :
- $k = 0$,
- T_0, c_0 donnés,
- k_0 = le nombre de cuissons avant une baisse de température, donné,
- calcul de $J(c_0)$.

Tant que critère de convergence non satisfait, faire

1. Générer aléatoirement un nouveau paramètre de contrôle c_{k+1},
2. Calculer $J(c_{k+1})$,
3. Test :
   - si $J(c_{k+1}) < J(c_k)$ alors
     conserver le point c_{k+1}
     $k \leftarrow k + 1$
   - si $J(c_{k+1}) > J(c_k)$ alors
     – définir la probabilité d'acceptation p grâce à l'équation (4.15),
     – générer une probabilité p',
     si $p' < p$ alors
     conserver le point c_{k+1},
     $k \leftarrow k + 1$,
     sinon, ne rien faire.
4. si $k \equiv 0[k_0]$ alors baisser la température

fin tant que.

On s'aperçoit facilement de l'importance de la donnée de la température initiale : plus celle-ci sera faible initialement, moins l'algorithme acceptera des configurations dégradées. Enfin, on voit clairement le caractère aléatoire présent dans cet algorithme, caractère totalement absent de la méthode du simplexe.

Dans notre cas, étant donné que nous souhaitons optimiser un grand nombre de paramètres, les probabilités de tomber dans un minimum local sont grandes. C'est pourquoi nous avons cherché à réduire dans un premier temps le nombre de paramètres à optimiser chez le poisson, puis nous avons choisi d'utiliser les méthodes du simplexe et du recuit-simulé pour optimiser une fonction objectif que nous définirons dans la section suivante.

4.3 Optimisation de la nage

Nous souhaitons à présent coupler le solveur de simulation de nage bidimensionnelle du poisson à un algorithme d'optimisation mathématique. Nous nous sommes concentrés

sur deux algorithmes non déterministes présentés précédemment : la méthode du simplexe et celle du recuit-simulé.

Étant donné le grand nombre de paramètres décrivant la forme et la nage du poisson, nous avons simplifié le problème en nous intéressant dans un premier temps uniquement à l'optimisation des paramètres de nage pour une géométrie de nageur donnée.

Dans ce cadre, nous avons choisi les paramètres de forme suivants :

- $\eta_c = -0.04$,
- $\alpha = 5$,
- $l = 1$.

Nous avons choisi de travailler à Reynolds 10^3, ce qui correspond à des poissons d'une longueur de corps proche de 3 cm et nageant à une vitesse d'une longueur de corps par seconde.

De plus, puisque nous nous concentrons sur un poisson de type thunniforme, nous faisons le choix raisonnable d'une longueur d'onde de la nage $\lambda = 1.25$. Il nous reste à présent trois paramètres de nage à optimiser :

- A l'amplitude de nage,
- c_1 ou c_2 les coefficients d'enveloppe de la nage,
- f la fréquence de la nage.

4.3.1 Optimisation des paramètres de nage

Nous avons souhaité progresser par ordre croissant de difficulté, par conséquent, nous réduirons dans une première étude à nouveau le nombre de paramètres à optimiser, puis nous considérerons les trois paramètres de nage comme paramètres de contrôle dans une seconde étude d'optimisation.

4.3.1.1 Deux paramètres de contrôle

Nous avons choisi de fixer l'amplitude de nage du poisson. Bainbridge [7] et Hunter *et al.* [56] ont montré que l'amplitude des oscillations de poissons tels que le maquereau, le saumon, le poisson rouge, la sardine ou encore le requin était d'environ 10% de la longueur du corps du poisson. Étant donné que nous nous intéressons à des poissons de bien plus petite taille, l'amplitude d'oscillation doit être plus élevée, tout en restant physiquement raisonnable. Nous avons donc fixé l'amplitude à 15% de la longueur du corps.

Introduisons à présent les objectifs que nous souhaitons atteindre.

Le premier objectif est la minimisation de la puissance moyenne (i.e. minimisation de la puissance instantanée) dépensée par le poisson sur une distance parcourue L_0. Nous imposons également une contrainte de vitesse horizontale u_{min}. Si nous n'imposons pas cette contrainte, l'algorithme d'optimisation risque de tendre vers la solution d'un poisson nageant à vitesse nulle, pendant un temps infini et dépensant une puissance instantanée nulle. Ainsi, en posant T_0 et T_1 respectivement le temps au départ et le temps à l'arrivée en

L_0 du poisson, alors la fonction objectif de minimisation de la puissance moyenne s'écrit :

$$\mathcal{J}_1 = \frac{1}{T_1 - T_0} \int_{T_0}^{T_1} \mathcal{P}(t)\,\mathrm{d}t + K\chi_u(\widehat{u}, u_{min}) + \sum_i \frac{\alpha_i}{2} c_i^2,$$

où,
$\mathcal{P}(t)$ est la puissance instantanée dépensée par le poisson,
K est un paramètre positif fixé par l'utilisateur,
$\widehat{u}$ est la vitesse moyenne horizontale du poisson lors du parcours de la distance L_0,
u_{min} est la contrainte de vitesse horizontale,
$\{c_i\}$ est l'ensemble des paramètres de contrôle considérés,
$\{\alpha_i\}$ sont des coefficients positifs fixés par l'utilisateur,
χ_u est une fonction indicatrice sur la vitesse, définie ci-après.

On définit χ_u comme :

$$\chi_u = \begin{cases} 1 \text{ si } |\widehat{u}| - |u_{min}| < 0, \\ 0 \text{ sinon.} \end{cases}$$

Dans toute la suite, on prend $u_{min} = 0.10$ et $L_0 = 3.5$.
Nous avons aussi choisi $K = 10^5$ et $\alpha_i = 10^{-6}, \quad \forall i$.

Ainsi, si le poisson nage à une vitesse moyenne inférieure en norme à la vitesse seuil, alors la fonction objectif deviendra très grande, ce qui rejettera ce jeu de paramètres dans l'algorithme d'optimisation. De plus le terme $\sum_i \frac{\alpha_i}{2} c_i^2$ assure que la fonction objectif est bien définie mathématiquement. Par ailleurs, les coefficients $\{\alpha_i\}$ sont pris très petits devant la puissance moyenne de sorte que la valeur de celle-ci soit l'élément prépondérant dans l'optimisation de la valeur de la fonctionnelle.

Puisque nous nous sommes réduits à deux paramètres de contrôle, nous avons d'abord testé l'optimisation en utilisant la méthode du simplexe.

Nous avons choisi les paramètres du simplexe suivants (se référer à §4.2.3 pour leur définition) :

- $\alpha = 1$,
- $\beta = 0.3$,
- $\gamma = 1.7$.

On impose ensuite des bornes aux paramètres de contrôle :

$$(c_1, f) \in [0; 1] \times [0.5; 3.5].$$

Enfin, le critère d'arrêt choisi se situe sur la longueur des arêtes du simplexe : si chaque longueur d'arête du simplexe est inférieure à $\tilde{l} = 10^{-4}$, on considère que l'algorithme a convergé. Notons que le choix de $\tilde{l}$ dépend des ensembles de définition des paramètres.

Nous avons également testé la convergence en effectuant plusieurs tests d'optimisation à simplexes initiaux différents. Les résultats de cette première optimisation sont rassemblés dans la table 4.2.

Simplexes initiaux		Résultats finaux			
c_1	f	c_1	f	$\mathcal{J}_1$	$\widehat{u}$
1 0.5 0	2 1 3	0.960	0.580	0.00476	-0.176
1 0.5 0	1 3 2	0.955	0.574	0.00470	-0.174
0.75 0.25 0.5	3.5 2.5 1.5	0.320	0.683	0.00501	-0.177
0.75 0.25 0.5	2.5 1.5 3.5	0.240	0.680	0.00470	-0.171

TABLE 4.2 – Optimisation par la méthode du simplexe de la puissance moyenne dépensée par le poisson sur une distance L_0.

À travers la table 4.2, on s'aperçoit que toutes les optimisations ont convergé vers une puissance moyenne minimale du même ordre de grandeur. Le paramètre de contrôle de la fréquence tend également vers une fréquence faible située entre 0.5 et 0.7. Tous les algorithmes n'ont cependant pas convergés vers le même coefficient c_1 bien que les valeurs de la fonctionnelle soient proches. Cela nous suggère deux hypothèses possibles. La première est l'existence de plusieurs minima locaux dans lesquels les différents simplexes ont convergé. La seconde, et celle qui nous semble la plus probable, est une faible dépendance de la fonctionnelle au paramètre c_1 à basse fréquence. Nous penchons vers cette hypothèse du fait que les vitesses moyennes des poissons optimisés sont très semblables malgré les paramètres de nage très différents.

Pour nous en convaincre, nous avons calculé une carte de valeurs. Nous avons donc estimé les valeurs de la puissance moyenne pour $(c_1, f) \in [0; 1] \times [0.5; 3.5]$, sur un maillage de taille 10×10. La carte est représentée en figure 4.13 et on y observe bien la faible variation de la puissance moyenne à basses fréquences pour les différentes valeurs de c_1.

Nous avons également vérifié que nous obtenions les mêmes zones de convergence des paramètres de contrôle, et de la fonctionnelle $\mathcal{J}_1$ pour d'autres paramètres du simplexe. Nous avons donc choisi

- $\beta' = 0.7$,
- $\gamma' = 1.3$.

Les résultats obtenus sont rassemblés dans la table 4.3 et sont très similaires à ceux de la table 4.2.

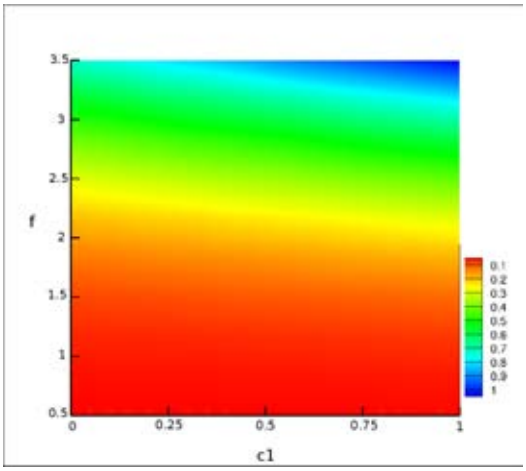


FIGURE 4.13 – Carte représentant les valeurs de $\mathcal{J}_1$ en fonction des paramètres c_1 et f.

Simplexes initiaux		Résultats finaux			
c_1	f	c_1	f	$\mathcal{J}_1$	$\widehat{u}$
1 0.5 0	2 1 3	0.935	0.580	0.00480	-0.176
1 0.5 0	1 3 2	0.835	0.560	0.00410	-0.159
0.75 0.25 0.5	3.5 2.5 1.5	0.107	0.697	0.00500	-0.175
0.75 0.25 0.5	2.5 1.5 3.5	0.402	0.622	0.00408	-0.156

TABLE 4.3 – Optimisation par la méthode du simplexe de la puissance moyenne dépensée par le poisson sur une distance L_0 pour un second jeu de paramètres.

De plus, nous avons comparé les résultats obtenus par la méthode du simplexe avec ceux résultant de l'algorithme du recuit-simulé.

Les paramètres du recuit simulé que nous avons choisis sont les suivants. Nous avons considéré une température initiale $T_{emp0} = 1$ pour laquelle nous avons choisi une loi de décroissance géométrique de coefficient 0.99. La température est abaissée chaque fois qu'une meilleure solution est trouvée. Nous avons choisi ces paramètres afin que l'algorithme ait la possibilité de parcourir le domaine de définition des paramètres sans être piégé prématurément dans un minimum local. En outre, le critère d'arrêt choisi s'applique sur le nombre d'itérations. On considère que s'il s'est écoulé 100 itérations sans que l'algorithme ait trouvé une meilleure solution alors l'algorithme a convergé vers la

solution optimale. Nous avons également effectué plusieurs algorithmes du recuit simulé pour différents points de départ afin d'observer l'influence de la condition initiale sur la solution. Les résultats sont rassemblés dans la table 4.4.

Points initiaux		Résultats finaux			
c_1	f	c_1	f	$\mathcal{J}_1$	$\widehat{u}$
1	1	0.26	0.55	0.00483	-0.171
0	3	0.42	0.54	0.00490	-0.172
0.5	2	0.47	0.53	0.00492	-0.174
0.875	3.5	0.34	0.54	0.00485	-0.171

TABLE 4.4 – Optimisation par la méthode du recuit-simulé de la puissance moyenne dépensée par le poisson sur une distance L_0.

On constate que les zones de convergence pour les paramètres de contrôle et la fonctionnelle sont similaires pour les deux méthodes. Par ailleurs, les zones de convergence obtenues par la méthode du recuit-simulé sont plus fines mais ne comprennent pas la meilleure valeur obtenue par le simplexe qui reste néanmoins proche de celles obtenues par le recuit-simulé.

Le second objectif que nous avons souhaité atteindre est très semblable au premier. Nous avons cherché à minimiser l'énergie totale dépensée par le poisson pour parcourir la distance L_0, en conservant toujours la contrainte de vitesse seuil. La seconde fonction objectif $\mathcal{J}_2$ s'écrit donc :

$$\mathcal{J}_2 = \int_{T_0}^{T_1} \mathcal{P}(t)\,\mathrm{d}t + K\chi_u(\widehat{u}, u_{min}) + \sum_i \frac{\alpha_i}{2} c_i^2.$$

Cette fonctionnelle a été introduite car ses applications sont très intéressantes au niveau ingénierie sous-marine, rendant ainsi son contexte d'applications plus réaliste. En effet, en minimisant l'énergie totale dépensée par le poisson sur une distance donnée, cela permet aux ingénieurs par exemple de connaître la loi de nage à donner à un robot sous-marin pour qu'il consomme le moins de carburant possible.

Nous avons utilisé les paramètres suivants [10] :
- $\alpha = 1$,
- $\beta = 0.3$,

10. Nous garderons ce jeu de paramètres pour toutes les autres optimisations utilisant la méthode du simplexe.

- $\gamma = 1.7$.

Les résultats obtenus sont rassemblés dans la table 4.5.

Simplexes initiaux		Résultats finaux			
c_1	f	c_1	f	$\mathcal{J}_2$	$\widehat{u}$
1 0.5 0	2 1 3	0.28	0.50	0.0693	-0.110
1 0.5 0	1 3 2	0.01	0.52	0.0673	-0.105
0.75 0.25 0.5	3.5 2.5 1.5	0.00	0.50	0.0646	-0.100
0.75 0.25 0.5	2.5 1.5 3.5	0.46	0.51	0.0735	-0.12

TABLE 4.5 – Optimisation par la méthode du simplexe de l'énergie totale dépensée par le poisson sur une distance L_0.

La direction d'une fréquence optimale apparaît ici clairement, il s'agit de la fréquence minimale admissible. Le poisson dépensant le moins d'énergie apparaît comme celui se déplaçant à la plus faible vitesse, ce qui est cohérent. Le paramètre de nage c_1 semble avoir dans ce cas une importance non négligeable sur la valeur de la fonctionnelle puisque l'on constate que les plus faibles valeurs de c_1, couplées à une basse fréquence ont fourni les plus faibles valeurs de la fonctionnelle Par conséquent, les paramètres optimaux sont : $c_1 = 0$ et $f = 0.5$, ce qui crée un poisson à nage extrêmement lente, dépensant ainsi le moins d'énergie au cours de sa nage. Nous avons enfin conforté ces résultats avec l'établissement d'une carte de valeurs $\mathcal{J}_2(c_1, f)$. Cette carte est représentée en figure 4.14.

Pour notre troisième objectif nous nous sommes intéressés à l'index propulsif I_p du poisson. En suivant la définition de [13], on définit l'index propulsif du poisson comme le rapport inverse entre la puissance $\mathcal{P}_{sps}$ fournie par le poisson autopropulsé pour parcourir une distance L_0, sur la puissance $\mathcal{P}_{tow}$ nécessaire pour tirer le poisson à la vitesse $\widehat{u}$ du poisson autopropulsé. Autrement dit,

$$I_p = \frac{\mathcal{P}_{tow}}{\mathcal{P}_{sps}}. \tag{4.16}$$

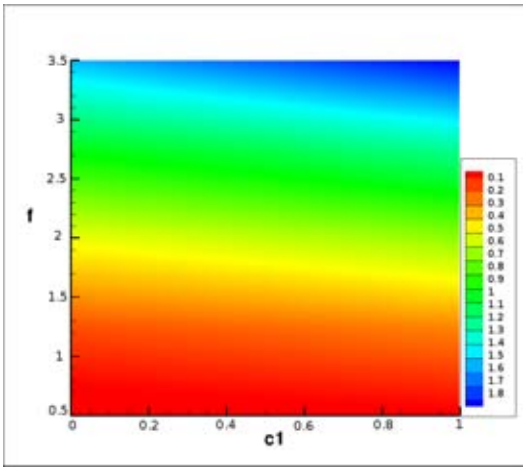


FIGURE 4.14 – Carte représentant les valeurs de $\mathcal{J}_2$ en fonction des paramètres c_1 et f.

On cherche alors à maximiser I_p afin de tendre vers la réduction de trainée. En effet, plus I_p sera grand, plus la trainée rencontrée par le poisson autopropulsé sera faible, ce qui est également intéressant pour des ingénieurs en termes de consommation de carburant.

Notre troisième fonction objectif s'écrit donc :

$$\mathcal{J}_3 = -I_p + K\chi_u\left(\widehat{u}, u_{min}\right) + \sum_i \frac{\alpha_i}{2} c_i^2.$$

La méthode du simplexe nous fournit les résultats rassemblés dans la table 4.6.

Simplexes initiaux		Résultats finaux			
c_1	f	c_1	f	$\lvert\mathcal{J}_3\rvert$	$\widehat{u}$
1 0.5 0	2 1 3	0.94	3.19	0.420	-1.67
1 0.5 0	1 3 2	0.98	3.06	0.418	-1.60
0.75 0.25 0.5	3.5 2.5 1.5	0.93	3.16	0.417	-1.64
0.75 0.25 0.5	2.5 1.5 3.5	0.78	3.16	0.417	-1.58

TABLE 4.6 – Optimisation par la méthode du simplexe de l'index propulsif du poisson sur une distance L_0.

On observe l'apparition d'une direction de convergence en c_1 et en f. Avec une géo-

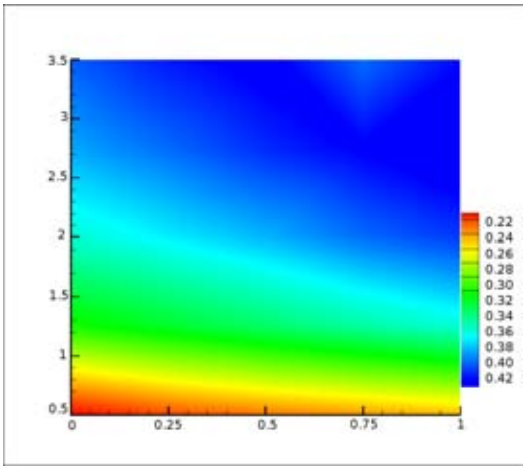


FIGURE 4.15 – Carte représentant les valeurs de $|\mathcal{J}_3|$ en fonction des paramètres c_1 et f.

métrie thunniforme, le maximum de l'index propulsif est donc atteint à haute fréquence avec un coefficient d'enveloppe c_1 proche de 1. Dans le but de valider une nouvelle fois ces résultats, nous avons établi une troisième carte $\mathcal{J}_3(c_1, f)$ (voir Figure 4.15). Celle-ci conforte les résultats obtenus.

L'index de propulsion défini dans l'équation (4.16) prend en considération la puissance requise pendant le régime transitoire (voir Figure 4.16). Toutefois, les conditions initiales en vitesse pour le poisson autopropulsé et pour le poisson tracté ne sont pas les mêmes : l'un possède initialement une vitesse nulle (le poisson autopropulsé) quand la vitesse du second est initialisée à $\widehat{u}$. Par conséquent, pour comparer des grandeurs identiques, on définit un second index propulsif $\tilde{I}_p$ tel que

$$\tilde{I}_p = \frac{\tilde{\mathcal{P}}_{tow}}{\tilde{\mathcal{P}}_{sps}},$$

où $\tilde{\mathcal{P}}_{tow}$ et $\tilde{\mathcal{P}}_{sps}$ sont respectivement les puissances dépensées par le poisson tiré et le poisson autopropulsé sur une distance $\tilde{L}_0$ en régime établi. Dans notre cas, on choisi $\tilde{L}_0 = 1$.

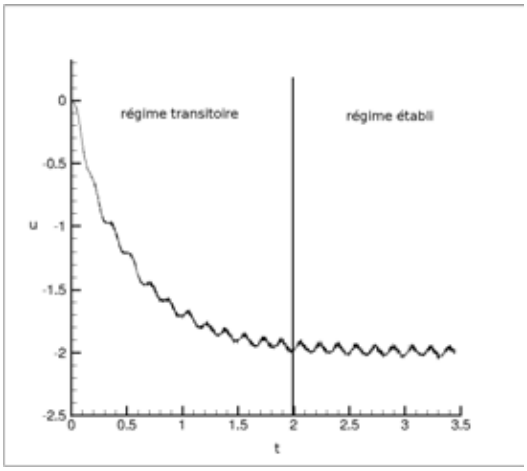


FIGURE 4.16 – Évolution en temps de la vitesse horizontale d'un poisson aux paramètres $\eta_c = -0.04$, $\alpha = 5$, $l = 1$, $\lambda = 1.25$, $A = 0.15$, $c_1 = 1$ et $f = 3$.

On pose alors notre quatrième fonction objectif :

$$\mathcal{J}_4 = -\tilde{I}_p + K\chi_u\left(\widehat{u}, u_{min}\right) + \sum_i \frac{\alpha_i}{2} c_i^2.$$

Dès lors, la méthode du simplexe nous fournit les résultats rassemblés dans la table 4.7.

Simplexes initiaux		Résultats finaux			
c_1	f	c_1	f	$\lvert\mathcal{J}_4\rvert$	$\widehat{u}$
1 0.5 0	2 1 3	0.80	3.11	0.581	-1.98
1 0.5 0	1 3 2	0.97	2.81	0.580	-1.79
0.75 0.25 0.5	3.5 2.5 1.5	0.50	3.32	0.573	-2.01
0.75 0.25 0.5	2.5 1.5 3.5	0.50	3.26	0.574	-1.97

TABLE 4.7 – Optimisation par la méthode du simplexe de l'index propulsif du poisson en régime établi sur une distance $\tilde{L}_0$.

On observe que les plus hautes valeurs d'index propulsif sont obtenues pour des paramètres c_1 proches de 1. Une direction de convergence pour f apparaît également bien qu'elle soit plus large que lors dès optimisations précédentes. Une dernière carte 4.17 est

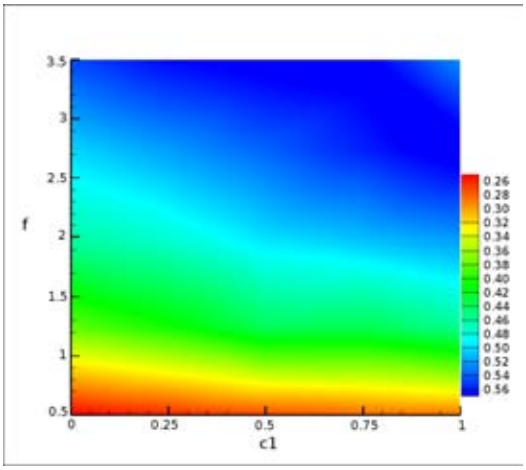


FIGURE 4.17 – Carte représentant les valeurs de $|\mathcal{J}_4|$ en fonction des paramètres c_1 et f.

établie $\mathcal{J}_4(c_1, f)$ est établie afin de confirmer les zones de convergence des paramètres de contrôle.

En conclusion, pour l'optimisation de deux paramètres de nage, nous observons que pour la minimisation de la puissance moyenne et de l'énergie dépensées, les paramètres de contrôle optimaux sont une fréquence faible pour les deux fonctionnelles, et un coefficient d'enveloppe c_1 faible pour la minimisation de l'énergie dépensée. Ces résultats sont cohérents puisque à faible fréquence, la nage devient plus lente, ce qui permet de minimiser l'énergie dépensée. De même, pour c_1 faible, le poisson courbera principalement sa nageoire caudale, réduisant ses déplacements verticaux, et diminuant l'énergie dépensée.

D'un autre côté, pour l'optimisation de l'index propulsif (en régime établi ou sur l'ensemble de la nage), il faut une fréquence et un coefficient d'enveloppe c_1 élevés. La fréquence élevée permet une meilleure propulsion du poisson, augmentant ainsi sa vitesse. L'index propulsif en régime établi est plus élevé que lorsqu'il est évalué sur l'ensemble de la nage puisque, en régime établi, on ne prend pas en compte la puissance nécessaire au poisson pour s'établir dans son régime permanent.

4.3.2 Trois paramètres de contrôle

Nous souhaitons maintenant rajouter l'amplitude de la nage comme paramètre de contrôle. Nous avons donc maintenant trois paramètres :

$$(A, c_1, f) \in [0.05; 0.25] \times [0; 1] \times [0.5; 3.5] .$$

Ayant mis en évidence les paramètres optimaux de fréquence et de coefficient de nage à amplitude donnée, nous nous plaçons autour de ces valeurs comme point de départ pour notre nouvelle optimisation.

Nous avons donc effectué l'optimisation des quatre fonctions objectifs précédemment définies en fonction des trois paramètres de nage, en utilisant la méthode du simplexe.

	Simplexes initiaux			Résultats finaux				
Fonctionnelle	A	c_1	f	A	c_1	f	$\lvert\mathcal{J}_i\rvert$	$\widehat{u}$
$\mathcal{J}_1$	0.185 0.17 0.20 0.15	0.47 0.43 0.4 0.5	0.6 0.5 0.45 0.65	0.099	0.51	0.65	0.00179	-0.10
$\mathcal{J}_2$	0.14 0.17 0.18 0.15	0 0.05 0.1 0.075	0.61 0.5 0.45 0.55	0.108	0.75	0.65	0.0639	-0.10
$\mathcal{J}_3$	0.14 0.17 0.18 0.15	1 0.9 0.8 0.95	3 3.2 3.1 3.15	0.128	0.93	3.49	0.425	-1.30
$\mathcal{J}_4$	0.14 0.17 0.18 0.15	1 0.9 0.8 0.95	2.8 2.9 3 3.1	0.110	1.00	3.34	0.585	-1.37

TABLE 4.8 – Optimisation par la méthode du simplexe des différentes fonctionnelles pour trois paramètres de contrôle de nage.

Les résultats obtenus sont rassemblés dans la table 4.8.

Nous observons qu'effectivement les directions de convergence en fréquence et pour le coefficient c_1 sont conservées, tout en étant légèrement supérieure aux optimisations précédentes à cause de la contrainte en vitesse. L'effet de cette contrainte de vitesse est ici particulièrement visible pour la fonctionnelle $\mathcal{J}_2$. Les valeurs des paramètres c_1 et f sont supérieures à celles obtenues pour une amplitude de 0.2 car pour des valeurs de f et c_1 faibles, la vitesse du poisson devient inférieure à la vitesse seuil pour $A = 10\%$.

Notons que nous avons également effectué des calculs d'optimisation en utilisant la méthode du recuit-simulé. Bien que cette approche nous ait montré de bons résultats lorsque nous avions deux paramètres de contrôle, nous n'avons pas obtenu la convergence lorsque l'optimisation de la fonctionnelle utilisait trois paramètres de contrôle. La raison vient certainement du choix des paramètres du recuit présentés précédemment, qui ne sont plus adaptés au cas de trois paramètres. Supposons que l'on garde le coefficient de décroissance de la température à 0.99, alors le critère de 100 itérations n'est plus suffisant car il ne laisse pas la possibilité à l'algorithme de parcourir le domaine. À coefficient de décroissance identique, il faudrait avoir un critère de 1000 itérations mais cela provoquerait des temps de calcul extrêmemnt longs. Par conséquent le coefficient de décroissance doit également être ajusté. Néanmoins, face à la rapidité des calculs en utilisant la méthode simplexe, nous avons préféré nous concentrer sur cette approche et laisser le soin de ce

	Simplexes initiaux			Résultats finaux				
Fonctionnelle	η_c	c_1	f	η_c	c_1	f	$\lvert\mathcal{J}_i\rvert$	$\widehat{u}$
$\mathcal{J}_1$	-0.04 -0.03 -0.035 0.025	0.47 0.43 0.4 0.5	0.6 0.5 0.45 0.65	-0.031	0.40	0.50	0.0039	-0.16
$\mathcal{J}_2$	-0.04 -0.03 -0.035 -0.025	0 0.05 0.1 0.075	0.61 0.5 0.45 0.55	-0.011	0.01	0.51	0.052	-0.11
$\mathcal{J}_3$	-0.04 -0.03 -0.035 -0.025	1 0.9 0.8 0.95	3 3.2 3.1 3.15	-0.01	0.90	3.05	0.554	-1.86
$\mathcal{J}_4$	-0.04 -0.03 -0.035 -0.025	1 0.9 0.8 0.95	2.8 2.9 3 3.1	-0.01	0.92	3.10	0.613	2.22

TABLE 4.9 – Optimisation par la méthode du simplexe des différentes fonctionnelles avec un paramètre de contrôle de forme et deux paramètres de contrôle de nage.

paramétrage à un futur utilisateur intéressé.

4.3.3 Optimisation de la forme et de la nage du poisson

Nous avons enfin effectué une optimisation à trois paramètres de contrôle sur la forme et la nage du poisson. Nous avons choisi pour se faire les paramètres suivants :

- l'épaisseur du poisson η_c,
- le coefficient d'enveloppe c_1,
- la fréquence de la nage f.

On fixe l'amplitude $A = 15\%$, comme précédemment. Les résultats finaux sont rassemblés dans la table 4.9

Nous pouvons observer qu'à amplitude fixée, dans le cas de la minimisation de l'énergie dépensée, l'algorithme tend vers un poisson de faible épaisseur, ce qui est cohérent.

4.4 Conclusion

Au cours de ce chapitre, nous nous sommes intéressés à l'optimisation de la nage et/ou de la forme des poissons.

Afin de comprendre quel type de poisson et quel type de nage nous avons choisi de modéliser, une présentation des différentes classes de nage ainsi que leurs différents modes associés a été effectuée. Nous avons choisi de nous placer dans le cadre de la nage

de type BCF où le poisson se sert principalement de son corps et de sa nageoire caudale pour avancer. Le mode BCF se caractérise par la propagation d'une onde rétrograde au sens de progression, le long du corps du poisson. Différentes catégories se distinguent ensuite suivant la proportion du corps traversée par l'onde.

Dans un second temps, nous avons présenté les différents types d'optimisation mathématique. Après avoir introduit les éléments fondamentaux à tout problème d'optimisation, deux classes de méthodes ont été présentées : les méthodes déterministes et les méthodes non déterministes. Les méthodes déterministes se basent sur l'évaluation du gradient de la fonction objectif en fonction des différents paramètres de contrôle. L'algorithme d'optimisation va ensuite suivre une direction de descente définie grâce au gradient afin de trouver un minimum de la fonctionnelle. Néanmoins, ces méthodes ne permettent pas de sortir d'un minimum local. Nous avons donc introduit deux méthodes non déterministes : la méthode du simplexe et celle du recuit-simulé. La méthode du simplexe crée un simplexe initial qui se déplacera dans le domaine des paramètres de contrôle par des opérations de reflexion, contraction ou expansion afin de se diriger vers le minimum de la fonctionnelle. Le recuit simulé se déplace aléatoirement dans le domaine des paramètres vers les valeurs optimales de la fonctionnelle. Toutefois, grâce à l'introduction d'une variable aléatoire de probabilité, cet algorithme garde la possibilité de sortir d'un minimum local en acceptant des solutions moins bonnes que celles déjà calculée.

Dans notre étude d'optimisation, nous avons choisi d'utiliser des méthodes non déterministes. Nous nous sommes tout d'abord concentrés sur l'optimisation de la nage d'un poisson donné. Nous avons choisi un poisson de géométrie type thunniforme. Ce choix nous a permis de fixer la longueur d'onde de la nage. De plus, souhaitant à nouveau réduire le nombre de paramètres, nous avons fixé l'amplitude de la nage. Nous obtenions donc deux paramètres à optimiser : l'enveloppe et la fréquence de la nage. Quatre fonctions objectifs ont été ensuite définies : la minimisation de la puissance moyenne (ou puissance instantanée), la minimisation de l'énergie dépensée, la maximisation de l'index propulsif défini sur l'ensemble de la nage et la maximisation de l'index propulsif en régime établi. Pour chaque fonctionnelle, nous avons obtenu des directions de convergence explicites. Ces directions ont été confirmées en restant similaires pour différents simplexes initiaux. De plus, nous avons confirmé ses résultats pour d'autres paramètres du simplexe et par un calcul de recuit simulé sur la minimisation de la puissance dépensée. Nous avons ainsi pu constater que la minimisation de la puissance moyenne et de l'énergie dépensée requerraient des fréquences de nage et coefficient d'enveloppe c_1 faibles. *A contrario*, la maximisation des index aboutit sur des fréquence et coefficient d'enveloppe c_1 forts.

Nous avons ensuite ajouté le paramètre de contrôle de l'amplitude de la nage. Pour cette optimisation à trois paramètres, nous nous sommes concentrés sur le simplexe. En partant de simplexes initiaux centrés sur les paramètres optimisés évalués précédemment, nous avons pu constater que les paramètres de fréquence et le coefficient d'enveloppe restaient inchangés dans la majorité des cas, ce à quoi nous nous attendions grâce à l'établissement de cartes de valeurs de la fonctionnelle. La seule étude où la direction de convergence des paramètres a été modifiée est due à la contrainte de vitesse choisie. Enfin, l'amplitude de nage optimisée est de l'ordre de 10% de la longueur du corps, ce qui est

cohérent avec la réalité observée.

Finalement, nous avons optimisé la forme et la nage en considérant les trois paramètres d'épaisseur du poisson, de la fréquence et du coefficient d'enveloppe c_1. L'optimisation a abouti sur les mêmes paramètres optimisés de fréquence et d'enveloppe. L'épaisseur du poisson optimisée varie quant à elle suivant l'objectif fixé. Par exemple, le poisson le plus fin est celui qui dépensera le moins d'énergie au cours de la nage, ce qui est raisonnable.

Par conséquent, nous avons réussi ici à coupler un solveur fluide d'interactions fluide-structure avec un algorithme d'optimisation mathématique. De plus, nous avons de bonnes convergences, cohérentes suivant les points de départ de l'optimisation et les phénomènes réalistes observés.

Chapitre 5

Simulation tridimensionnelle de la nage

Sommaire

5.1 L'outil Squelette 120
5.1.1 Définitions 120
5.1.2 La détermination numérique du Squelette 121
5.2 Simulation tridimensionnelle de la nage 124
5.2.1 Génération d'une géométrie tridimensionnelle approchée 124
5.2.2 Simulation 3D de la nage 131
5.3 Génération d'une géométrie tridimensionnelle réaliste 134
5.3.1 Redressement, reconstruction des fonctions Level Set et mise à l'échelle 136
5.3.2 Construction de la fonction Level Set 3D 137
5.4 Conclusion 146

Avec le souhait de toujours nous rapprocher davantage de la réalité, nous avons développé deux outils : l'un nous fournissant une reconstruction approchée d'un poisson dont nous sommes capables de simuler la nage, et le second nous permettant de reconstruire une géométrie réaliste 3D.

Nous avons décidé pour cela d'utiliser l'objet mathématique du *Squelette*. En effet, les collaborations issues de l'ANR Carpeinter nous ont permis de constater la praticité de cet outil lors de la reconstruction de vaisseaux sanguins ou de canaux.

Nous nous sommes concentrés sur des poissons de type thunniforme ne présentant pas de nageoires significatives puisque nous cherchons à modéliser un même type de poisson que dans l'étude 2D, mais avec des proportions issues de la réalité.

Concernant la simulation de la nage, notre point de départ sera deux photos du poisson : une vue de profil et une vue de dessus. La vue de face, qui correspond à une coupe transversale du poisson sera approchée par une ellipse. En partant des deux images, nous en extrairons les valeurs des hauteurs et largeurs le long du corps du poisson, qui correspondront aux différentes valeurs des petits et grands axes des ellipses. Pouvant alors suivre lagrangiennement le corps du poisson ainsi reconstitué, nous pourrons simuler sa nage, de la même manière qu'en 2D.

Cette reconstitution étant une approximation, nous avons souhaité être plus réaliste encore. Nous avons donc développé un second outil qui reconstitue la géométrie du poisson à partir des trois photos : les vues de dessus et de profil comme précédemment, auxquelles s'ajoute la vue de face réaliste. À partir de ces trois données, une reconstitution réaliste de la géométrie du poisson peut être effectuée.

Par conséquent, après une présentation de l'outil Squelette, nous verrons comment nous avons simulé la nage d'un poisson dont la géométrie est approchée. Nous l'applique-

rons au cas d'une espèce de la famille des maquereaux : le chinchard. Puis nous reconstituerons complètement une géométrie tridimensionnelle d'un poisson : celle du maquereau commun.

5.1 L'outil Squelette

5.1.1 Définitions

La notion de squelette fut introduite par Blum en 1967 [16]. Blum considère le contour d'une prairie à partir duquel se propage un feu à vitesse constante et vers l'intérieur de la prairie. Il nomme alors *Squelette* l'ensemble des points où les fronts du feu se rencontrent. Ces points sont également appelés "points d'extinction" étant donné que le feu y meurt, ne pouvant plus se propager face à la terre déjà brulée.

Blum introduit ensuite une définition plus mathématique du squelette, définition qui constitue une formulation très utilisée de nos jours. Ainsi, après introduction de la fonction distance au contour d'un objet, il définit le squelette de cet objet comme l'ensemble des points singuliers du gradient de la fonction distance.

Une autre approche pour définir le squelette d'un objet est celle de Calabi [21] qui utilise la notion de *Boule maximale*, qu'il convient de définir ici.

Définition 5.1 : une Boule maximale

Soit Ω_d un ensemble borné ouvert de $\mathbb{R}^d$, $d \geqslant 1$. Soit $\mathcal{B}(\mathbf{x_0}, r(\mathbf{x_0}))$ une boule ouverte de $\mathbb{R}^d$ centrée en $\mathbf{x_0}$ et de rayon $r(\mathbf{x_0})$.

On dit que $\mathcal{B}$ est une boule maximale de Ω_d si et seulement si, quelle que soit $\mathcal{B}'$ une boule ouverte de $\mathbb{R}^d$,

$$\mathcal{B} \subseteq \mathcal{B}' \Rightarrow \mathcal{B} = \mathcal{B}'.$$

Calabi définit alors le squelette d'un objet comme l'ensemble formé des centres de ses boules maximales. Il démontre de plus que les notions de centres des boules maximales et points d'extinction sont équivalentes (voir Figure 5.1 [1]).

On remarque alors que si $\mathbf{x}$ est un point du squelette d'un objet donné alors la boule maximale $\mathcal{B}(\mathbf{x}, r(\mathbf{x}))$ de cet objet est tangente au contour de l'objet en deux points au moins.

Notons également que si, à chaque point du squelette on attache le rayon de la boule maximale qui lui est associée, alors les notions de squelette et d'axe médian [2] sont similaires.

Le squelette est un outil très utilisé pour la reconnaissance et la description des formes [97]. Il constitue donc un outil particulièrement intéressant pour la manipulation des formes [52] [53] dans des domaines tels que l'animation. Il trouve aussi de grandes

1. Cette image est issue du cours de squelettisation de Marc Neveu, université de Bourgogne (Dijon)
2. l'axe médian d'une surface est l'ensemble des centres des boules qui coïncident avec le contour de la surface en plus d'un point.

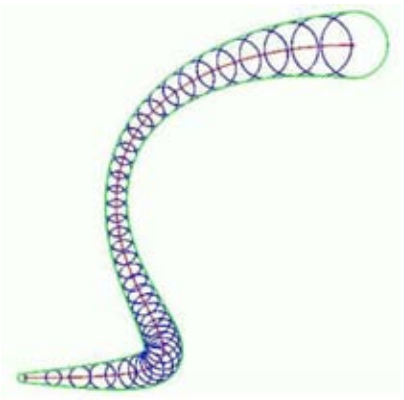


FIGURE 5.1 – Définition du squelette d'un objet par la notion de boules maximales.

applications dans l'imagerie médicale, pour la reconstruction numérique 3D du système sanguin [42] [93] par exemple.

Le squelette d'un objet satisfait, en théorie[3], les conditions suivantes :

1. le squelette doit être homotope à la géométrie de l'objet considéré. Autrement dit, le squelette conserve la topologie de l'objet qu'il représente. Par exemple, si l'objet est connexe alors son squelette l'est aussi.
2. le squelette doit être invariant par isométrie.
3. le squelette doit être réversible. Autrement dit, l'objet doit pouvoir être complètement reconstruit à partir de son squelette.

5.1.2 La détermination numérique du Squelette

Il existe un grand nombre de méthodes pour déterminer le squelette d'un objet. Nous ne les présenterons pas toutes ici mais trois d'entre elles le seront brièvement afin de voir la variété des approches possibles. Les méthodes que nous présenterons sont :

1. les méthodes d'amincissement,
2. les méthodes de type propagation de feu de prairie,
3. les méthodes basées sur la fonction distance.

5.1.2.1 Les méthodes d'amincissement

Cette approche ne se base sur aucune des idées qui ont pu être citées jusqu'à présent. Il s'agit d'une méthode itérative qui consiste à approximer le squelette d'un objet par des amincissements successifs de la géométrie [49] [14].

En effet, on cherche ici à conserver la topologie initiale de l'objet en parcourant les points de son contour et en enlevant ceux dont la suppression ne modifie pas sa topologie.

Cette technique, facilement parallélisable, fut utilisée entre autre par Tozaki *et al.* [102] pour extraire les axes médians de vaisseaux sanguins.

3. Ces conditions ne sont pas toujours respectées dans le domaine discret.

5.1.2.2 Les méthodes de type propagation de feu de prairie

Cette méthode garde le même principe que l'approche de Blum [16] : on se place initialement sur le contour de l'objet et on considère un feu se déplaçant vers l'intérieur de l'objet à vitesse constante.

Les nouveaux fronts de feu sont déterminés à chaque instant [4] et les intersections de ces fronts constituent l'ensemble des points d'extinction formant le squelette [119].

Cette approche présente néanmoins quelques défauts. En effet, cette méthode est difficilement parallélisable. De plus, le squelette obtenu est très sensible aux variations locales de la géométrie de l'objet [119].

5.1.2.3 Les méthodes basées sur la fonction distance

Cette méthode se base sur l'utilisation d'une fonction distance, signée ou non. On choisira ici le cas d'une fonction distance signée, correspondant à la fonction Level Set ϕ utilisée depuis le début de notre étude.

Dans cette approche, comme il l'a été mentionné précédemment, les points constituant le squelette correspondent aux points singuliers de la fonction $\nabla\phi$, en l'occurrence, aux minima locaux de $|\nabla\phi|$.

De surcroît, on peut également faire le lien avec la méthode identifiant le squelette à l'ensemble des centres des boules maximales. Effectivement, dès que les points singuliers de $\nabla\phi$ sont identifiés, la valeur de la fonction Level Set ϕ en ces points nous donne le rayon de leur boule maximale associée.

Cette méthode fut utilisée entre autre pour la reconstruction de canaux et la simulation d'écoulements par Galusinski et Nguyen [41].

Dans notre étude, étant donné que nous travaillons avec l'outil Level Set, il a été naturel d'utiliser cette approche basée sur la fonction distance.

5.1.2.4 Remarque importante

Le point pouvant poser problème dans notre cas, *a priori*, se situe dans la définition, sur chaque vue du poisson, de son contour. En effet, les photos du poisson étant issues de la réalité, la forme du poisson (et par conséquent la fonction Level Set) est inconnue *a priori*.

Pour résoudre ce problème nous avons utilisé un logiciel développé par Cedric Galusinski et Christian Nguyen [41] permettant de définir la fonction distance signée d'un objet quelconque à partir d'une image.

Dans un premier temps, le logiciel permet à l'utilisateur d'améliorer la qualité de l'image d'entrée grâce à des opérations sur les pixels : par des augmentations locales ou globales de contrastes, ainsi que par l'utilisation de filtres (linéraires et non linéaires) servant à éliminer des bruits parasites sur l'image. Si ces outils s'avèrent insuffisants, l'utilisateur peut également utiliser des outils basés sur des EDP type Ginzburg-Landau capables entre autre de reconnecter une interface interrompue localement.

4. Cela peut être facilement effectué avec l'utilisation de la fonction Level Set par exemple.

Il en résulte à cet instant, pour l'utilisateur, une image binaire de l'objet (par exemple : l'objet est noir, le reste du domaine est blanc) (voir Figure 5.2).

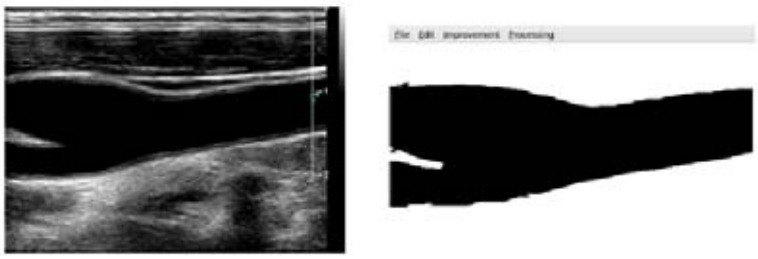


FIGURE 5.2 – Traitement d'une image doppler d'une carotide par le logiciel : à gauche l'image originale, à droite l'image après traitement [41].

Afin maintenant de déterminer la fonction Level Set de l'objet, ils utilisent une technique appelée "*floodfile*".

Pour cela, on définit une vitesse normale de propagation V telle que $V = 1$ sur les pixels noirs (i.e. dans l'objet) et $V = 0$ sur les pixels blancs (i.e. partout ailleurs).

Ensuite, l'utilisateur identifie une région initiale interne à l'objet (voir Figure 5.3). Cette région croît à la vitesse V jusqu'aux bords de l'objet grâce à la résolution de l'équation eikonale suivante :

$$\begin{cases} |\nabla T(\mathbf{x})| = \dfrac{1}{V(\mathbf{x})} & \forall \mathbf{x} \in \Omega, \\ T(\mathbf{x}) = 0 & \forall \mathbf{x} \in \Omega_0 \subset \Omega, \end{cases}$$

où $\begin{cases} \Omega \text{ est le domaine consitué par l'image,} \\ \Omega_0 \text{ est la région initiale déterminée par l'utilisateur,} \\ T \text{ correspond au temps d'arrivée de la région grandissante à l'interface de l'objet,} \\ V \text{ est la vitesse normale à la région.} \end{cases}$

L'objet en question correspond alors aux points où le temps d'arrivée T est fini.



FIGURE 5.3 – Délimitation d'une région initiale circulaire par l'utilisateur [41].

Une fois que les points intérieurs de l'objet sont identifiés, on peut connaître le contour de l'objet $\partial\Omega_0$. Dès que $\partial\Omega_0$ est identifié, la fonction Level Set est déterminée par la résolution de l'équation eikonale :

$$\begin{cases} |\nabla\phi(\mathbf{x})| = 1, & \forall \mathbf{x} \in \Omega, \\ \phi(\mathbf{x}) = 0 & \forall \mathbf{x} \in \partial\Omega_0. \end{cases}$$

Enfin, ϕ étant connue, le squelette peut enfin être calculé comme il l'a été décrit précedemment (voir Figure 5.4).

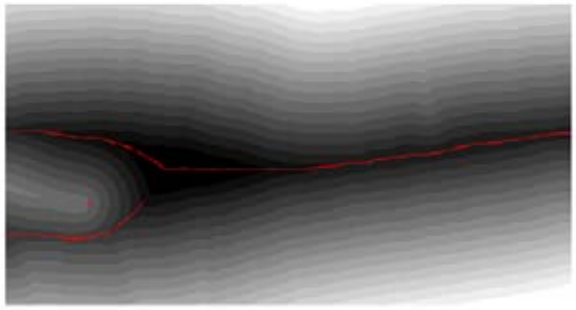


FIGURE 5.4 – Résultat final obtenu grâce au logiciel : la fonction Level Set et le squelette [41]. Les courbes de niveaux de gris représentent la fonction Level Set, et l'ensemble de points rouges, le squelette.

5.2 Simulation tridimensionnelle de la nage

Nous allons ici présenter une reconstruction approchée du poisson qui nous permettra de simuler sa nage. Cette reconstruction est dite approchée car elle approxime la coupe transversale d'un poisson (i.e. la vue de face) par une ellipse.

Nous allons tout d'abord présenter comment nous recréons ce poisson gâce aux données fournies par le logiciel de Cédric Galusinski et Christian Nguyen se basant sur deux photos du poisson. Puis nous observerons l'application d'une loi de nage sur le poisson reconstruit.

5.2.1 Génération d'une géométrie tridimensionnelle approchée

Désormais, dans toute la suite de ce manuscrit, on définira un squelette $\mathcal{S}$ d'une géométrie Ω_0 comme :

$$\mathcal{S} = \{(\mathbf{x}_0, r_0) \text{ tel que } \mathcal{B}(\mathbf{x}_0, r_0) \text{ est une boule maximale de } \Omega_0\}.$$

Grâce au logiciel développé par Cedric Galusinski et Christian Nguyen, nous sommes dorénavant capables d'obtenir la fonction Level Set d'une forme quelconque.

L'idée pour reconstruire un nageur, en l'occurrence de type poisson, en trois dimensions est de partir des vues de profil et de dessus, d'en extraire l'ensemble des

hauteurs et largeurs respectivement, le long du corps du poisson. Puis, on reconstitue le poisson en approximant chaque coupe transversale par une ellipse dont les valeurs des petit et grand axes correspondent à la largeur et la hauteur de la coupe correspondante.

Nous illustrerons cette étude en l'appliquant au cas d'un chinchard.

La première hypothèse[5] importante pour notre reconstruction est la suivante :

Hypothèse 1 : Hypothèse d'uniformisation
On suppose que la vue de face du poisson constitue une représentation de toute coupe transversale du poisson, à un facteur homothétique près.

La vue de face est ici approximée par une ellipse. Les facteurs homothétiques seront issus des vues de profil et de dessus. En effet, grâce à l'introduction d'une notion, le "*squelette médian*", nous verrons comment nous sommes en mesure d'obtenir les largeur et hauteur de chaque coupe du poisson.

Par conséquent, la reconstruction 3D du poisson consiste à représenter ce dernier comme la juxtaposition de l'ensemble de ses coupes transversales.

Notons que les deux vues initiales du poisson en entrée du logiciel sont deux photos prises manuellement. Les photos ne sont donc pas, *a priori*, à la même échelle, ni nécessairement droites. Il nous faudra donc redresser et mettre à la même échelle les données.

Finalement, la construction 3D du poisson s'articule autour des étapes suivantes :

1. Redressement des vues,
2. Mise à l'échelle des données,
3. Construction des marqueurs lagrangiens sur le contour 3D du poisson.

5.2.1.1 Redressement des vues

Notre point de départ est deux fichiers issus du logiciel de Cédric Galusinski et Christian Nguyen nous fournissant les fonctions Level Set pour les vues de profil et de dessus, et les coordonnées de tous les points de leur squelette avec le rayon de la boule maximale qui lui est associée (voir Figure 5.5).

Avant de chercher à définir une géométrie 3D, nous devons rendre les données compatibles entre elles. En effet, comme il l'a été dit plus tôt, les photos du poisson sont prises manuellement, il n'est donc pas garanti que les vues soient droites ou à la même échelle.

Interviennent alors notre seconde et notre troisième hypothèses importantes pour la reconstruction :

5. Au cours de ce chapitre, nous allons énoncer trois hypothèses. Celles-ci seront formulées dans le cas général où la vue de face n'est pas approchée par une ellipse. Ceci est effectué dans le but d'éviter toute redite dans la section suivante. Elles seront bien entendu vérifiées également dans le cas de l'approximation par l'ellipse.

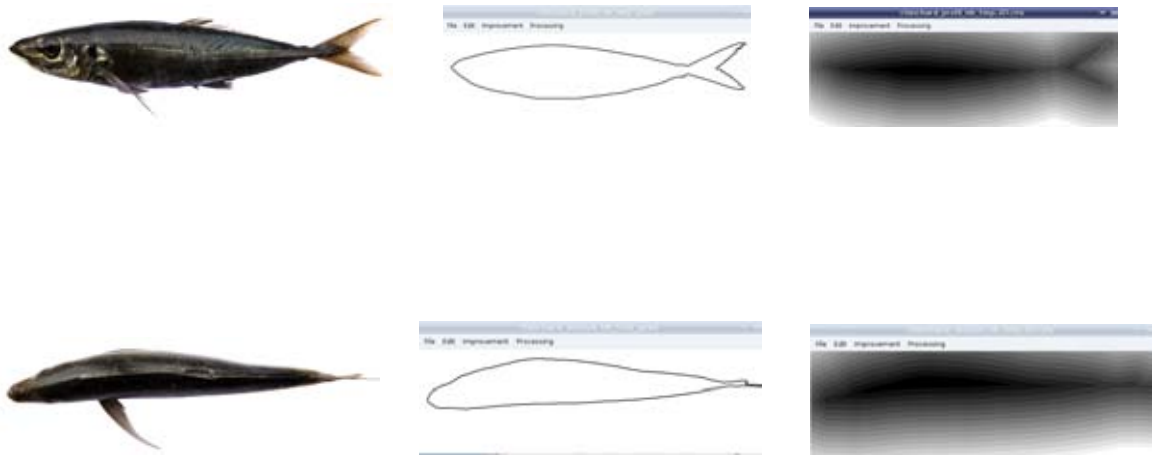


FIGURE 5.5 – Représentation d'un chinchard : de haut en bas, profil puis dessus ; de gauche à droite, photo, contour et Level Set (ces deux derniers étant fournis par le logiciel).

Hypothèse 2 : Hypothèse de symétrie

On suppose que les vues de dessus et de face possèdent un axe de symétrie.

Le poisson vu de dessus est symétrique par rapport à son arête dorsale et le poisson vu de face est symétrique par rapport à l'axe vertical passant au milieu de ses deux yeux.

Hypothèse 3 : Hypothèse de faible déformation

On suppose que, bien que les vues ne soient pas droites, elles ne sont pas non plus "très" déformées.

On suppose que l'axe de symétrie du poisson sur la vue de dessus est "proche" de l'axe horizontal ; de même, on suppose que sur la vue de face, l'axe de symétrie de la coupe est "proche" de l'axe vertical.

Cette dernière hypothèse est primordiale dans notre procédure de redressement de la vue de dessus.

La phase de redressement constitue en l'introduction d'un nouvel outil : *le squelette médian redressé.*

Définition du squelette médian redressé

Afin de définir le squelette médian d'une vue, une étape préliminaire est nécessaire. Il s'agit du stockage de toutes les épaisseurs verticales du poisson le long de sa géométrie.

Introduisons au préalable quelques notations (qui seront utiles également pour la suite). On notera :

- Nx_d, Ny_d et ϕ_d le nombre de colonnes, de lignes et la fonction Level Set respectivement sur la vue de dessus,
- Nx_p, Nz_p et ϕ_p le nombre de colonnes, de lignes et la fonction Level Set respectivement sur la vue de profil.

De même, on pose
- Lx_d et Ly_d les dimensions respectivement horizontale et verticale sur la vue de dessus,
- Lx_p et Lz_p les dimensions respectivement horizontale et verticale sur la vue de profil.

Considérons la vue de dessus et la fonction Level Set qui lui est associée.
L'algorithme de stokage des épaisseurs du poisson est le suivant :

Algorithme 5.1 : Stockage des épaisseurs sur la vue de dessus

Tant que $j \leqslant Ny_d$, faire
 Tant que $i \leqslant Nx_d$, faire
  si le point (i, j) est dans le poisson (i.e. $\phi_d(i, j) \leqslant 0$), alors
  - parcourir tous les points de la colonne j jusqu'à sortir du poisson, (*La Longueur parcourue L_i nous donne la largeur du poisson*)
  - stocker le couple $\left(x_i, \frac{L_i}{2}\right)$.

  $i \leftarrow i + 1$
 fin tant que.
$j \leftarrow j + 1$
fin tant que.

On s'aperçoit ici de l'importance de l'hypothèse de faible déformation puisque, en l'occurrence, on confond la largeur réelle du poisson en x_i avec la largeur projetée sur l'axe verticale $y = y_j$.

Une fois que l'ensemble des couples $\tilde{\mathcal{L}} = \{\left(x_i, \frac{L_i}{2}\right)\}$ est déterminé, on introduit la notion de distance verticale d_y.

Définition 5.2 : La distance[6] verticale d_y

Soient $\mathbf{x} = (x, y)$ et $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$ deux points de $\mathbb{R}^2$.

On définit la distance verticale d_y de $\mathbf{x}$ à $\tilde{\mathbf{x}}$ comme :

$$d_y(\mathbf{x}, \tilde{\mathbf{x}}) = |y - \tilde{y}|.$$

Nous pouvons à présent introduire la notion de squelette médian.

6. Notons que, en dépit de sa dénomination, d_y ne correspond pas à la définition mathématique d'une distance dans $\mathbb{R}^2$ puisqu'elle ne vérifie pas la propriété de séparation :

$$\forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^2, d_y(\mathbf{x}, \tilde{\mathbf{x}}) = 0 \not\Leftrightarrow \mathbf{x} = \tilde{\mathbf{x}}.$$

Définition 5.3 : Le squelette médian

Le squelette médian d'un ensemble Ω_0 correspond à l'ensemble des centres des boules maximales de Ω_0 définies par rapport à la distance verticale d_y.

Redresser la vue de dessus du poisson revient alors à définir l'ensemble des points du squelette médian du dessus de la manière suivante :

Définition 5.4 : Le squelette médian redressé de la vue de dessus

Soit $\mathcal{S}_{m,d}$ le squelette médian redressé de la vue de dessus du poisson.

$$\mathcal{S}_{m,d} = \left\{ \left((x,y), \frac{L}{2} \right) \text{ tels que } y = \frac{Ly_d}{2}, \left(x, \frac{L}{2} \right) \in \tilde{\mathcal{L}} \right\}.$$

La définition du squelette médian de la vue de profil diffère de la définition du squelette médian de la vue de dessus étant donné que le squelette médian de la vue de profil ne sera pas redressé car on suppose que la vue de profil du poisson n'est **pas** symétrique.

Pour définir le squelette médian de la vue de profil, on va définir la fonction Level Set verticale $\tilde{\phi}_p$ de la vue de profil, qui correspond à la fonction distance signée par rapport à la distance verticale d_y.

L'algorithme pour définir $\tilde{\phi}_p$ est le suivant :

Algorithme 5.2 : Définition de la fonction distance signée verticale $\tilde{\phi}_p$

Tant que $j \leqslant Nz_p$, faire
Tant que $i \leqslant Nx_p$, faire

1. si le point (i,j) est en dehors du poisson (i.e. $\tilde{\phi}_p(i,j) > 0$), alors
   $\tilde{\phi}_p(i,j) = 1$,
2. si le point (i,j) est dans le poisson (i.e. $\tilde{\phi}_p(i,j) \leqslant 0$), alors
   - on définit $\tilde{\phi}_{bas}$ et $\tilde{\phi}_{haut}$ correspondant à la distance verticale du point (i,j) au contour du profil du poisson qui lui est inférieur et supérieur respectivement,
   - $\phi_p(i,j) = -min\left(\tilde{\phi}_{bas}, \tilde{\phi}_{haut}\right)$.

$i \leftarrow i+1$
fin tant que.
$j \leftarrow j+1$
fin tant que.

Une fois $\tilde{\phi}_p$ construite, les points du squelette médian de la vue de profil correspondent aux points intérieurs au profil et minima de la fonction $|\nabla\tilde{\phi}_p|$. De plus, $\tilde{\phi}_p(i,j)$ nous donne le rayon de la boule verticale qui lui est associée, par rapport à la distance verticale d_y.

Dès lors, le squelette médian $\mathcal{S}_{m,p}$ de la vue de profil est défini par :

$$\mathcal{S}_{m,p} = \left\{ \left((x,z), \frac{h}{2} \right) \text{ tels que } (x,z) \text{ sont les minima locaux de } |\nabla \tilde{\phi}_p| \text{ et } \frac{h}{2} = |\tilde{\phi}_p(x,z)| \right\}.$$

On peut de plus définir la fonction Level Set associée à la vue redressée de dessus de la manière suivante.

L'ensemble $S_{0,d}$ des points formant la courbe de niveau zéro de la vue de dessus est :

$$S_{0,d} = \left\{ (x,y) \text{ tels que } \exists L \in \mathbb{R}^+, \left(x, \frac{L}{2} \right) \in \tilde{\mathcal{L}} \text{ et } y = \frac{Ly_d \pm L}{2} \right\}.$$

On définit à présent une première fonction Levet Set (temporairement erronée) de la vue du dessus par :

$$\phi_d(x,y) = \begin{cases} 0 & \text{si } (x,y) \in S_{0,d}, \\ -1 & \text{si } \exists L \in \mathbb{R}^+, \left(x, \frac{L}{2} \right) \in \tilde{\mathcal{L}} \text{ et } \frac{Ly_d - L}{2} \leqslant y \leqslant \frac{Ly_d + L}{2}, \\ +1 & \text{sinon.} \end{cases}$$

Pour définir correctement la fonction Level Set ϕ_d, on résout enfin l'équation eikonale :

$$\begin{cases} |\nabla \phi_d| = 1, \\ \phi_d(x,y) = 0, \qquad \forall (x,y) \in S_{0,d}. \end{cases}$$

On obtient enfin la fonction Level Set ϕ_d associée à la vue de dessus redressée (voir Figure 5.6).



FIGURE 5.6 – Chinchard avant (à gauche) et après (à droite) redressement.

5.2.1.2 Mise à l'échelle

Rien ne garantissant que la vue de profil et la vue de dessus soient à la même échelle, il nous faut déterminer le facteur homothétique reliant les échelles propres à chaque vue. Pour se faire, on utilisera les valeurs de longueur totale du poisson sur chacune des vues.

Sur la vue de dessus, la longueur du poisson $l_{p,d}$ est donnée par :

$$l_{p,d} = |x_{queue,d} - x_{nez,d}|,$$

où $x_{queue,d}$ et $x_{nez,d}$ désignent l'abscisse des points extrêmes du poisson sur la vue de dessus, au niveau de la queue et du nez respectivement, dont la donnée est directement

donnée par le squelette médian redressé de la vue de dessus.

Sur la vue de profil, on définit de la même manière $x_{queue,p}$ et $x_{nez,p}$, et donc $l_{p,p}$ la longueur du corps du poisson.

On choisit par convention la vue de profil comme vue de référence[7] et on définit α le coefficient d'homothétie des échelles :

$$\alpha = \frac{l_{p,p}}{l_{p,d}}.$$

Enfin, pour mettre les deux vues à la même échelle, on pose :

$$\begin{cases} Lx_d \leftarrow Lx_d * \alpha, \\ Ly_d \leftarrow Ly_d * \alpha, \\ \forall \left((x,y), \dfrac{L}{2} \right) \in \mathcal{S}_{m,d}, \qquad \begin{cases} x \leftarrow x_{nez,p} + (x - x_{nez,d}) * \alpha \\ y \leftarrow y * \alpha \\ L \leftarrow L * \alpha. \end{cases} \end{cases}$$

Ainsi, les abscisses du nez et de la queue sont identiques sur les deux vues.
On aboutit donc à des squelettes médians de profil et de dessus compatibles.

5.2.1.3 Définition des marqueurs lagrangiens le long du contour du poisson 3D

Il nous faut à présent reconstruire l'ensemble des ellipses le long du corps du poisson.

On connaît la position des centres des ellipses : leurs abscisses x et ordonnées y sont l'ensemble des points appartenant au squelette médian de la vue de dessus redressée, et la dernière composante z est récupérée dans le squelette médian de la vue de profil pour l'abscisse x correspondante.
Une fois que les points sont identifiés dans les deux squelettes médians, alors le demi-petit axe et le demi grand-axe correspondent respectivement à la largeur stockée correspondante dans le squelette médian de la vue de dessus, et à la hauteur stockée dans le squelette médian de la vue de profil.

Connaissant maintenant l'expression analytique d'une ellipse, on peut repérer le nombre de marqueurs lagrangiens désiré sur chaque ellipse.

Remarque :

Cette construction pose un problème lorsque l'on se situe au niveau du départ en fourche de la queue. En effet, à ce niveau, pour une abscisse x donnée correspond plusieurs valeurs de la composante z. À ce moment là, on fait l'approximation suivante.

7. le choix de la vue de dessus comme vue de référence est équivalent.



FIGURE 5.7 – Représentation du chinchard reconstruit non déformé en haut, déformé en bas à gauche (vue de dessus) et avec maillage lagrangien en bas à droite.

Soient z_1 et z_2 les deux composantes correspondantes associées aux hauteurs respectives h_1 et h_2.

On considère que le centre de l'ellipse se situe en $z = \dfrac{z_1 + z_2}{2}$ et la hauteur associée est $h = \dfrac{|z_1| + |z_2| + h_1 + h_2}{2}$.

Cela revient à éliminer la fourche et considérer la queue comme pleine.

Nous avons appliqué cette reconstruction au cas du chinchard. La géométrie obtenue est représentée dans la figure 5.7.

5.2.2 Simulation 3D de la nage

Une fois en possession de l'ensemble des marqueurs lagrangiens du poisson reconstruit, la simulation de la nage en 3D n'est que peu différente de la simulation en 2D. En effet, la loi de nage s'applique à l'ensemble formé de tous les centres des ellipses qui constitue la nouvelle colonne vertébrale du poisson, puis la géométrie est reconstituée en replaçant les marqueurs sur les ellipses comme il l'a été décrit précedemment.

L'unique différence intervient au niveau du solveur qui, étant devenu tridimensionnel devient plus coûteux en temps de calculs. Effectivement, l'utilisation d'une grille cartésienne, de schémas de type différences finies, et de la méthode IPC rendent le passage à la simulation tridimensionnelle très facile au niveau de l'implémentation. Toutefois, les temps de calculs peuvent quant à eux devenir particuliérement longs. Par conséquent, le code est parallélisé grâce à l'utilisation de la librairie PETSc[8]. Le nombre important

8. www.mcs.anl.gov/petsc

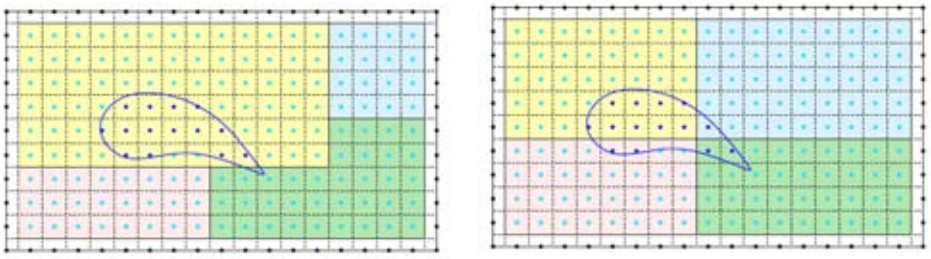


FIGURE 5.8 – Partitionnements de domaine : inefficace à gauche, efficace à droite

de points de maillage pour les simulations tridimensionnelles rendent les calculs séquentiels particulièrement longs. Le recours à la parallélisation est alors courante pour obtenir des temps de calculs raisonnables. Durant cette procédure, les calculs sur l'ensemble des noeuds sont répartis sur plusieurs processeurs, contrairement aux calculs séquentiels où tous les calculs sont effectués sur un unique processeur.

Bien entendu, afin que la parallélisation soit efficace, les noeuds doivent être correctement répartis sur chacun des processeurs, sinon la parallélisation ne sera pas au maximum de sa potentialité d'optimisation du temps de calcul (voir Figure 5.8).

Nous avons donc choisi de simuler la nage du chinchard. Le chinchard étant de la famille des maquereaux, il est un poisson thunniforme : seule la nageoire caudale se déforme. De plus il est particulièrement rapide. Toutes ces caractéristiques se traduisent par les paramètres de nage suivants :

$$\begin{cases} A = 0.2, \\ c_1 = 0, \\ c_2 = 1, \\ \lambda = 1, \\ f = 3. \end{cases}$$

Les résultats des simulations pour ces paramètres de nage sont exposés dans la figure 5.9 correspondant à un chinchard de 20cm nageant une longueur de corps par seconde.

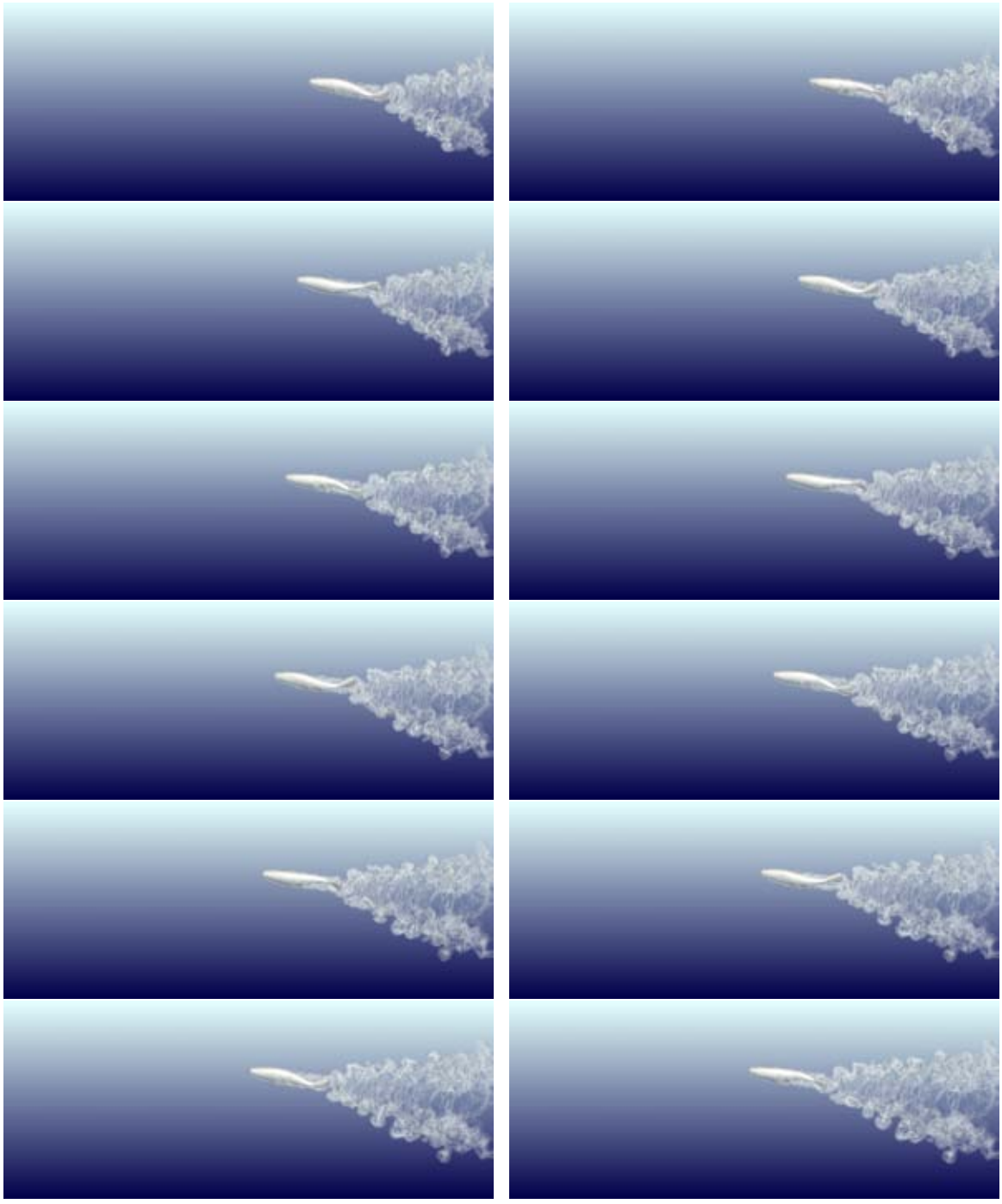


FIGURE 5.9 – Simulation de la nage du chinchard reconstruit

On observe que les sillages provoqués par la nage du poisson sont beaucoup plus espacés dans la simulation 3D par rapport aux simulations 2D où ceux-ci sont très resserrés derrière le poisson (voir Figure 5.2.2). Cette forme en "V" qui apparaît en 3D pourrait expliquer la forme des bancs d'oiseaux volant dans le ciel par exemple. En effet, les oiseaux en arrière profiteraient des recirculations d'air provoquées par l'oiseau en amont, diminuant les efforts qu'ils devraient fournir pour voler.

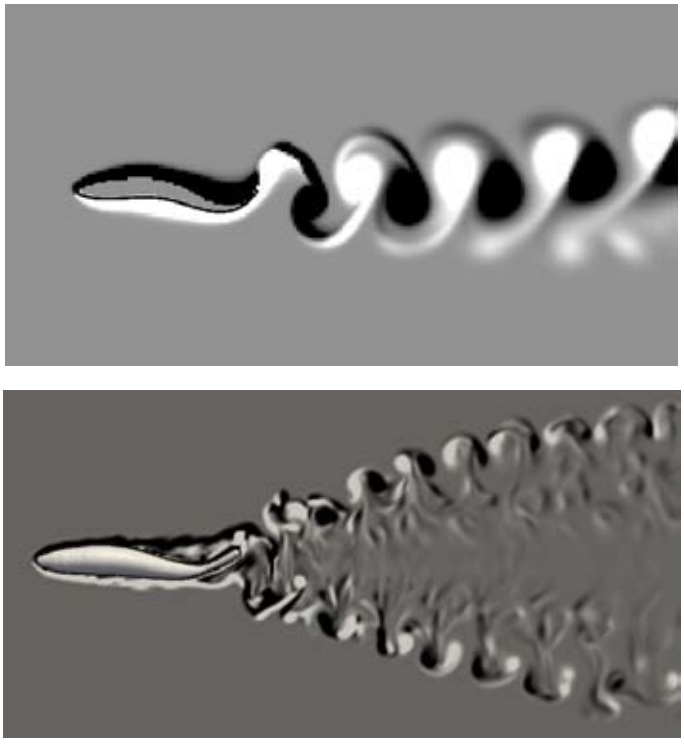


FIGURE 5.10 – Comparaison des sillages 2D et 3D d'un poisson : en haut, sillage 2D, en bas, vue de dessus du sillage 3D.

5.3 Génération d'une géométrie tridimensionnelle réaliste

Souhaitant nous rapprocher toujours plus de la réalité, nous avons décidé de prolonger notre travail vers une reconstruction réaliste. Pour se faire, nous allons utiliser les trois vues dont nous disposons. Par conséquent, nous n'approximerons plus la vue de face par une ellipse mais garderons celle issue de la photo.

Nous illustrerons cette reconstruction avec cette fois-ci le cas d'un maquereau commun.

Notre nouveau point de départ est donc les trois vues : profil, dessus, face et leur fonction Level Set associée obtenue grâce au logiciel de Cédric Galusinski et Christian Nguyen (Figure 5.11).

On suppose que les trois hypothèses formulées en section §5.2 sont vérifiées.

Notons néanmoins que l'hypothèse d'uniformisation impose que l'on ne modélisera pas des poissons présentant une nageoire quelconque significative. En effet, prenons par exemple un requin (voir Figure 5.12 [9]), la vue de face du requin présente les nageoires dorsale et pectorales et ne constitue pas une coupe quelconque représentative du requin. L'hypothèse 1 nous amènerait à reconstruire ces nageoires sur toute la longueur du corps

9. source : http ://kaaz.eu/requin/

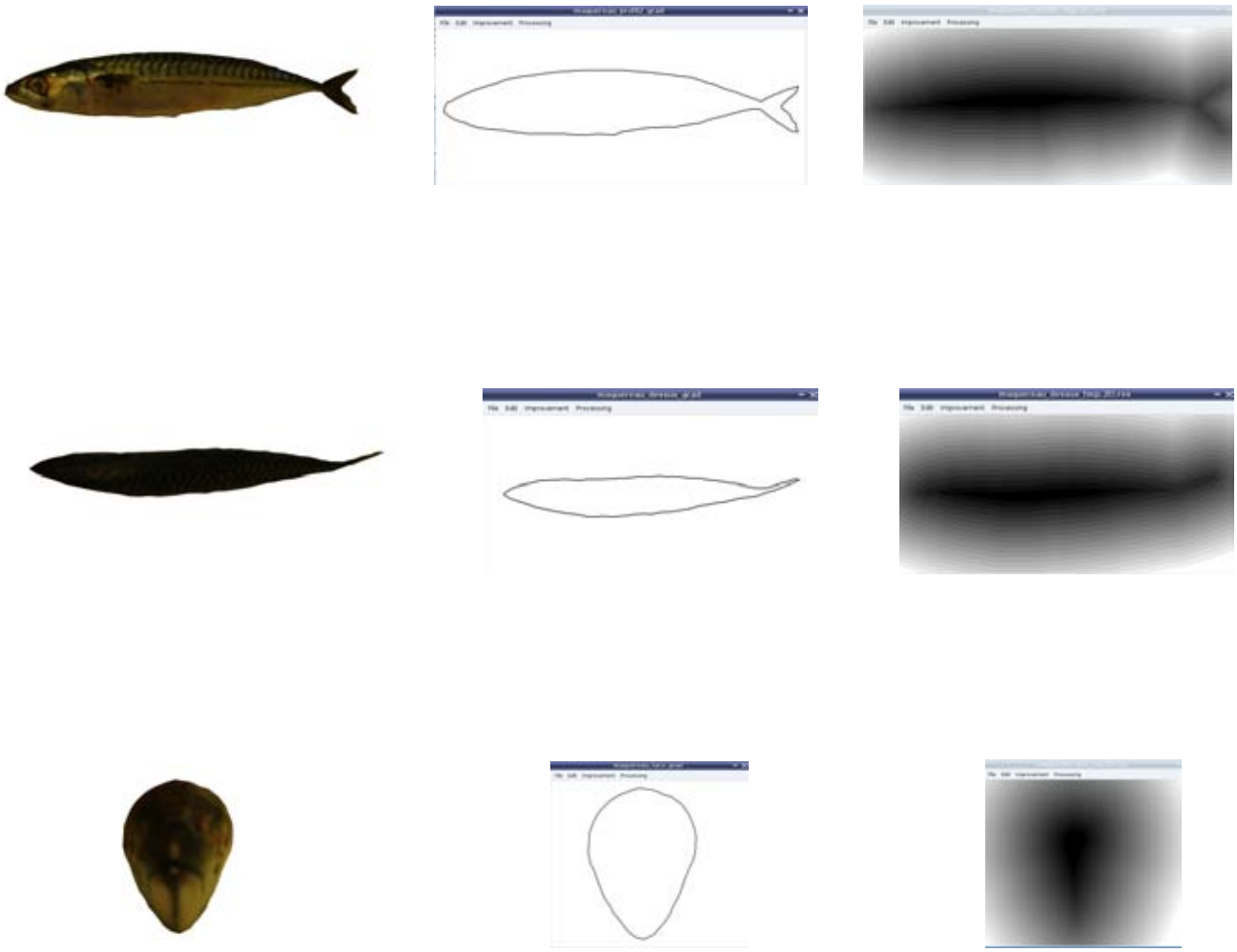


FIGURE 5.11 – Représentation d'un maquereau : de haut en bas, profil, dessus puis face ; de gauche à droite, photo, contour et Level Set (ces deux derniers étant fournis par le logiciel).

du requin, ce qui serait bien entendu erroné.



FIGURE 5.12 – Vue de face d'un requin blanc : à gauche l'image originale, à droite l'image binaire produite par le logiciel.

La démarche présentée dans §5.2.1 est conservée ici puisqu'on considère toujours initialement que les vues ne sont ni droites, ni à la même échelle.

Par conséquent, la construction 3D du poisson s'articule autour des étapes suivantes :

1. Redressement, reconstruction des fonctions Level Set et Mise à l'échelle des données,
2. Construction de la Level Set 3D.

5.3.1 Redressement, reconstruction des fonctions Level Set et mise à l'échelle

Cette étape est en tout point similaire à celle présentée en §5.2.1. La seule différence ici est le redressement supplémentaire de la vue de face.

La première étape de la reconstruction 3D est, similairement, le redressement de la vue de dessus et de la vue de face.

Nous ne rentrerons pas dans les détails pour cette étape car elle est identique à celle présentée en §5.2.1. Nous poserons simplement les éléments propres à la vue de face.

On introduit donc les notations supplémentaires suivantes :

- Ny_f, Nz_f et ϕ_f le nombre de colonnes, de lignes et la fonction Level Set respectivement sur la vue de face,
- Ly_f et Lz_f les dimensions respectivement horizontale et verticale sur la vue de face.

Notons que l'on suppose l'axe de symétrie de la vue de face verticale, le redressement de la vue de face s'effectuera donc sur la vue de face **transposée**.

Ainsi, on stoke l'ensemble des épaisseurs sur la vue de face transposée en créant l'ensemble $\tilde{l} = \{(z_i, l_i)\}$, ce qui nous permet de créer le squelette médian redressé de la vue de face transposée.

Définition 5.5 : Le squelette médian redressé de la vue de face transposée

Soit $\mathcal{S}_{m,f}$ le squelette médian redressé de la vue de face du poisson.

$$\mathcal{S}_{m,f} = \left\{ \left((z, y), \frac{l}{2} \right) \text{ tels que } y = \frac{Ly_f}{2}, \left(z, \frac{l}{2} \right) \in \tilde{l} \right\}$$

En outre, de la même manière que pour la vue de dessus, on définit la fonction Level Set ϕ_f associée à la vue de face redressée.

Ensuite, on effectue une mise à l'échelle entre les vues de profil et de dessus, en tout point similaire à celle présentée en §5.2.1.

On suppose donc avoir en main des vues de dessus et de face redressées ainsi que des vues de dessus et de profil à la même échelle (voir Figure 5.13). L'étape suivante consiste à définir la fonction Level Set 3D du poisson.

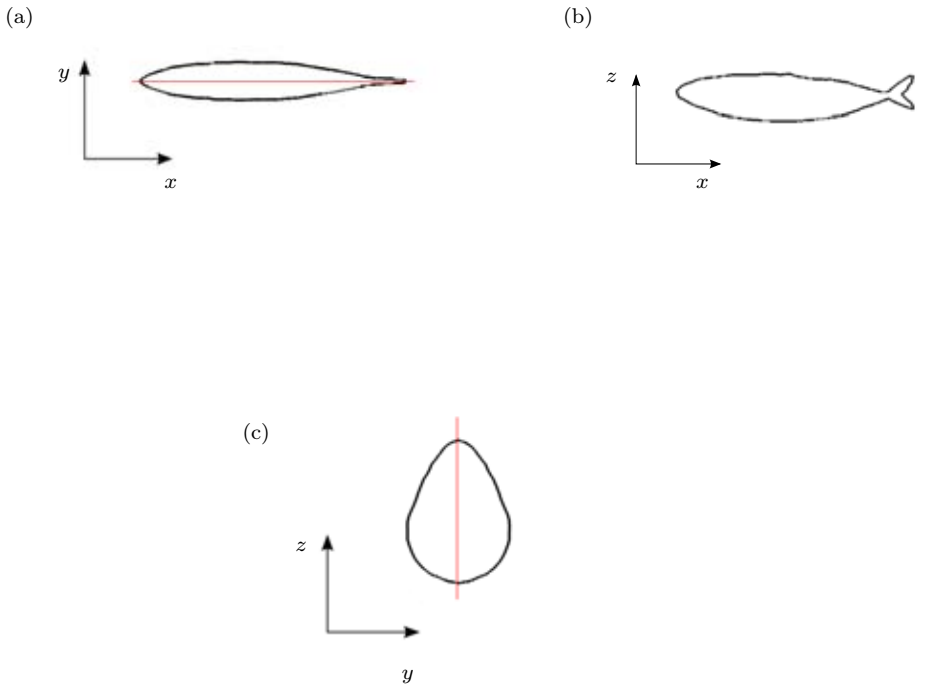


FIGURE 5.13 – Les trois vues du maquereau après redressement et mise à l'échelle : (a) vue de dessus, (b) vue de profil, (c) vue de face. Les axes rouges représentent les axes de symétrie.

5.3.2 Construction de la fonction Level Set 3D

La construction de la fonction Level Set 3D consiste à donner une expression de ϕ particulière suivant la position du point considéré. En partant du postulat $x_{nez} < x_{queue}$, on distinguera trois cas (voir Figure 5.14) :

soit $\mathbf{x} = (x, y, z)$ un point de l'espace approximé numériquement par $\mathbf{x}\,(i, j, k)$,

- cas 1 : $\mathbf{x}$ est devant le nez du poisson (i.e. $x < x_{nez}$),
- cas 2 : $\mathbf{x}$ est derrière la queue du poisson (i.e. $x > x_{queue}$),
- cas 3 : $\mathbf{x}$ est entre le nez et la queue (i.e. $x_{nez} \leqslant x \leqslant x_{queue}$).

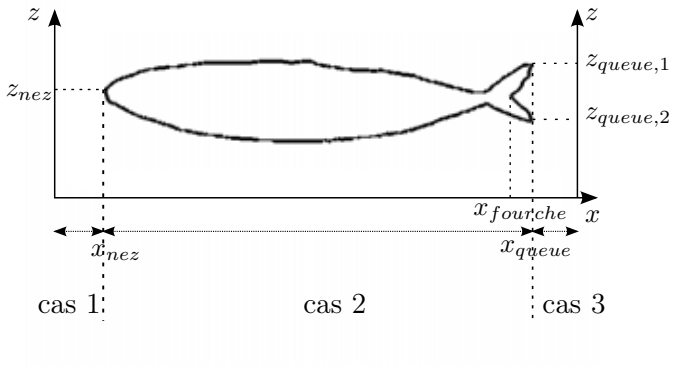


FIGURE 5.14 – Identification des différents cas pour la détermination de la Level Set 3D du poisson

5.3.2.1 Cas 1 : x est devant le nez du poisson

Dans ce cas, on approxime $\phi(i,j,k)$ par la distance de **x** au nez du poisson.

Grâce à la vue de dessus, on connaît x_{nez} et y_{nez}. En outre, le squelette médian de la vue de profil nous fournit z_{nez}. Par conséquent, pour tous les points situés devant le nez du poisson, on peut calculer ϕ :

$$\phi(i,j,k) = \sqrt{(x - x_{nez})^2 + (y - y_{nez})^2 + (z - z_{nez})^2}.$$

On considère pour l'instant, pour les cas 2 et 3, un poisson non déformé (on reviendra sur ce cas ultérieurement, lors de la présentation du cas d'un poisson déformé).

5.3.2.2 Cas 2 : x est derrière la queue

Dans ce cas, de la même manière que pour le cas 1, on approxime $\phi(i,j,k)$ par la distance du point considéré à l'extrémité de la queue.

x_{queue}, y_{queue} et z_{queue} nous sont donnés par les squelettes médians des vues de profil et de dessus. On a alors

$$\phi(i,j,k) = \sqrt{(x - x_{queue})^2 + (y - y_{queue})^2 + (z - z_{queue})^2}.$$

Notons que, dans le cas où il existe deux points extrémités au niveau de la queue (exemple en figure 5.14), alors on détermine la distance à chaque extrémité. Puis, on identifie $\phi(i,j,k)$ au minimum des deux distances.

5.3.2.3 Cas 3 : x est entre le nez et la queue

Certains poissons peuvent présenter une queue en forme de fourche. La manière de déterminer ϕ sera alors différente suivant si $\mathbf{x}$ se situe près du corps ou au niveau de la queue en fourche. Pour savoir dans quelle situation nous nous trouvons, il faut tout d'abord déterminer $x_{fourche}$ qui représente l'abscisse du point de départ de la queue en fourche (voir Figure 5.14).

$x_{fourche}$ correspond à l'abscisse du point du squelette médian de la vue de profil vérifiant les conditions suivantes :

Soit $\left((x,z),\frac{h}{2}\right)$ un point du squelette médian de la vue de profil.

$$\begin{cases} \phi_p(x,z) \leqslant 0, \\ \phi_p(x-\triangle x,z) < 0, \\ \phi_p(x+\triangle x,z) > 0. \end{cases}$$

Il existe en réalité trois points du squelette médian de la vue de profil vérifiant cette condition : le point de départ en fourche que nous cherchons à déterminer, et les deux points extrêmes de la queue sur chaque branche de la fourche.

Une fois ces trois points identifiés, $x_{fourche}$ correspond l'abscisse minimale des trois points.

Cas 3.1 : x se situe au niveau du corps

Pour connaître la distance de $\mathbf{x}$ au corps du poisson, il nous faut tout d'abord reconstruire la coupe transversale correspondante.

La coupe correspond à la fonction Level Set de la face, homothétisée et translatée dans le domaine 3D global.

Considérant la vue de face transposée du poisson, le squelette médian de la vue de face nous permet de connaître la largeur L_{max} et la hauteur h_{max} maximales du poisson (voir Figure 5.15).

Pour connaître les facteurs homothétiques correspondant à la coupe, il faut aller chercher dans les vues de profil et de dessus les valeurs de hauteur et largeur réelles.

Connaissant l'abscisse x où l'on se situe, il suffit d'aller chercher dans le squelette médian de la vue de profil la hauteur h réelle correspondante, et dans le squelette médian redressé de la vue de dessus, la largeur L réelle associée à l'abscisse x.

Une fois h et L connues, les facteurs homothétiques appliqués à la Level Set ϕ_f de la face et à son squelette sont $\frac{h}{h_{max}}$ dans la direction z, et $\frac{L}{L_{max}}$ dans la direction y.

Il reste désormais à savoir où se situe la coupe considérée dans le domaine global.

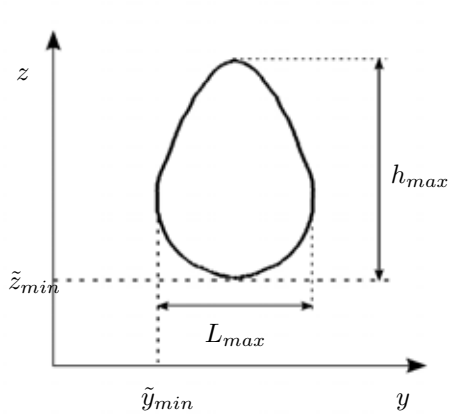


FIGURE 5.15 – Identification des différents paramètres sur la vue de face en vue des futures homothéties et translations.

Pour cela, on repère les coordonnées minimales de la fonction Level Set ϕ_f $(\tilde{y}_{min}, \tilde{z}_{min})$ (voir Figure 5.15) dans la vue de face homothétisée. Connaissant l'abscisse x, la vue de profil nous permet de connaître la position réelle z_0 de ce point, et la vue de dessus nous donne y_0 (voir Figure 5.16).

Dès lors, nous sommes en mesure de translater la fonction Level Set de la face, ainsi que son squelette médian, pour les positionner correctement dans le domaine global 3D.

Une fois la fonction Level Set ϕ_f de la coupe et son squelette médian correspondant déterminés, on peut connaître la distance de $\mathbf{x}$ au corps [10].

- si $\phi_f(y,z) > 0$ alors $\mathbf{x}$ est en dehors du poisson.
  On pose alors $z_{min} = z_0$, $z_{max} = z_0 + h$, $y_{min} = y_0$ et $y_{max} = y_0 + L$ (voir Figure 5.16).
  - si $z > z_{max}$, alors $\phi(i,j,k) = \sqrt{(z - z_{max})^2 + (y - y_{min})^2}$,
  - si $z < z_{min}$, alors $\phi(i,j,k) = \sqrt{(z - z_{min})^2 + (y - y_{min})^2}$,
  - sinon, on pose $\phi_1 = |y - y_{max}|$, $\phi_2 = |y - y_{min}|$ (voir Figure 5.17).
    D'où, $\phi(i,j,k) = min(\phi_1, \phi_2)$.

- si $\phi_f(y,z) \leqslant 0$ alors $\mathbf{x}$ est dans le corps du poisson.
  On repère alors $\left((\tilde{y},\tilde{z}),\tilde{l}\right)$ le point du squelette médian de la face (homothéisée et translatée) le plus proche de (y,z).
  On pose alors $\phi_1 = |y - (\tilde{y} + l)|$, $\phi_2 = |y - (\tilde{y} - l)|$. (voir Figure 5.17)
  Enfin, $\phi(i,j,k) = -min(\phi_1, \phi_2)$.

10. Notons l'approximation : dans le cas présent, nous allons supposer que la distance réelle du point au contour 3D du poisson est assimilable à sa distance verticale au contour.

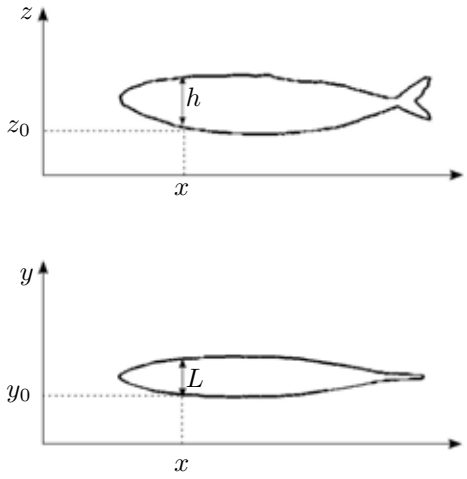


FIGURE 5.16 – Identification des hauteur et largeur réelles sur la vue de profil (en haut) et sur la vue de dessus redressée (en bas).

Cas 3.2 : x se situe au niveau de la queue

Dans le cas où **x** se situe au niveau de la queue, **x** aura plusieurs points du squelette correspondants à son abscisse, et par conséquent plusieurs coupes et coefficients homothétiques associés.

Pour savoir quelle coupe correspond à **x**, on construit les deux fonctions Level Set 3D au point (i, j, k) et on prend simplement la plus petite valeur trouvée.

Cas particulier : Construction d'un poisson déformé

Dans le cas où le poisson est déformé, le principe global de construction reste le même. Les modifications à apporter sont décrites ci-après.

Tout d'abord, les tests de position ne s'effectuent plus sur la coordonnée x de **x**. On définit un nouvel élément : l'abscisse de coupe x_{curv}.

Pour définir l'abscisse de coupe, on considère tout d'abord le squelette médian déformé de la vue de dessus [11]. On détermine alors le point du squelette médian déformé le plus proche du point **x**. Ce point correspond au projeté orthogonal de **x** sur l'axe médian redressé de la vue de dessus [12]. On notera x_{curv} l'abscisse curviligne de ce point

11. Nous ne décrivons pas ici la déformation du squelette médian du dessus car celle-ci correspond exactement au processus de déformation de l'axe médian du poisson 2D, présenté au chapitre 4.

12. Tous les points de l'espace devant le nez du poisson auront donc comme projeté orthogonal le nez du

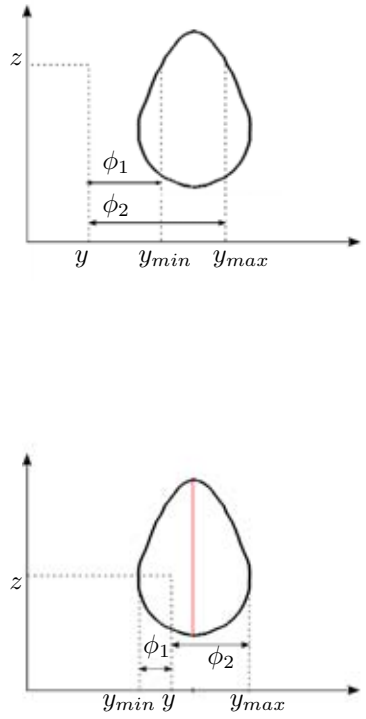


FIGURE 5.17 – Définition des différents éléments pour la détermination de la fonction Level Set 3D dans le cas où le point considéré est en dehors du poisson (en haut) ou à l'intérieur du poisson (en bas). L'ensemble des points rouges représente le squelette médian redressé de la coupe transversale.

le long du squelette médian déformé de la vue de dessus du poisson. x_{curv} nous indique dans quelle coupe transversale se situe $\mathbf{x}$ (voir Figure 5.18). En effet, x_{curv} correspond à l'abscisse de la coupe dans le cas d'un poisson non déformé. Il ne nous reste alors plus qu'à aller chercher dans la vue de profil et la vue de dessus non déformée les valeurs des hauteur et largeur réelles h et L comme il l'a été décrit précédemment, et de reconstruire la coupe transversale du poisson dans le plan normal au squelette médian déformé de la vue de dessus.

Dans le cas où x_{curv} correspond au point extrême de la queue du poisson, on suivra la démarche suivante :

1. On considère tout d'abord le plan $\mathcal{P}$ orthognal à la queue du poisson dans la vue de dessus déformée (voire Figure 5.19). On détermine alors la distance d de $\mathbf{x}$ au plan $\mathcal{P}$.
2. On introduit ensuite $\tilde{\mathbf{x}}$ la projection orthogonale de $\mathbf{x}$ sur $\mathcal{P}$. Puis, on détermine la distance $\tilde{d}$ de $\tilde{\mathbf{x}}$ à l'extrémité de la queue du poisson, d'une manière identique à celle présentée pour le cas 2.

poisson même. De même, les points situés derrière la queue du poisson auront comme projeté orthogonal le point extrême de la queue du poisson.

3. Enfin, on pose $\phi(i,j,k) = \sqrt{\left(d^2 + \tilde{d}^2\right)}$.

Avant d'exposer le résultat final pour la reconstruction 3D d'un poisson dans le cas d'un maquereau, faisons une dernière remarque.

Il est possible que les fonctions Level Set issues du logiciel de Cedric Galusinski et Christian Nguyen soient bruitées. Dans de tels cas, la géométrie 3D reconstruite sera altérée. Pour éviter cela, on applique un filtre passe bas de type gaussien sur les vues bruitées afin d'obtenir une image plus lisse.

Finalement, après toutes ces étapes, les Figures 5.20 et 5.21 fournissent le résultat final suivant dans le cas du maquereau.

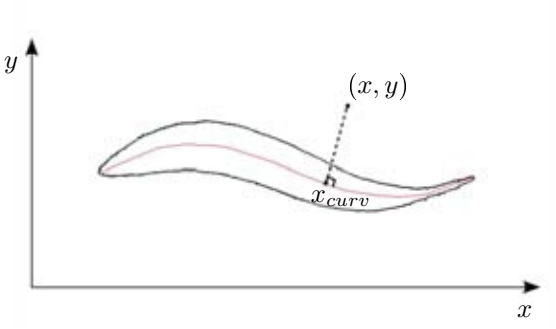


FIGURE 5.18 – Définition de x_{curv} sur la vue de dessus déformée du poisson. L'ensemble de points rouges représente le squelette médian déformé de la vue de dessus.

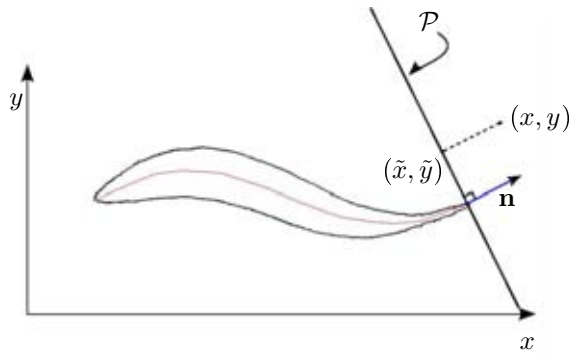


FIGURE 5.19 – Projection du point courant dans le plan orthogonal à la queue du poisson. **n** représente le vecteur normal au plan $\mathcal{P}$.

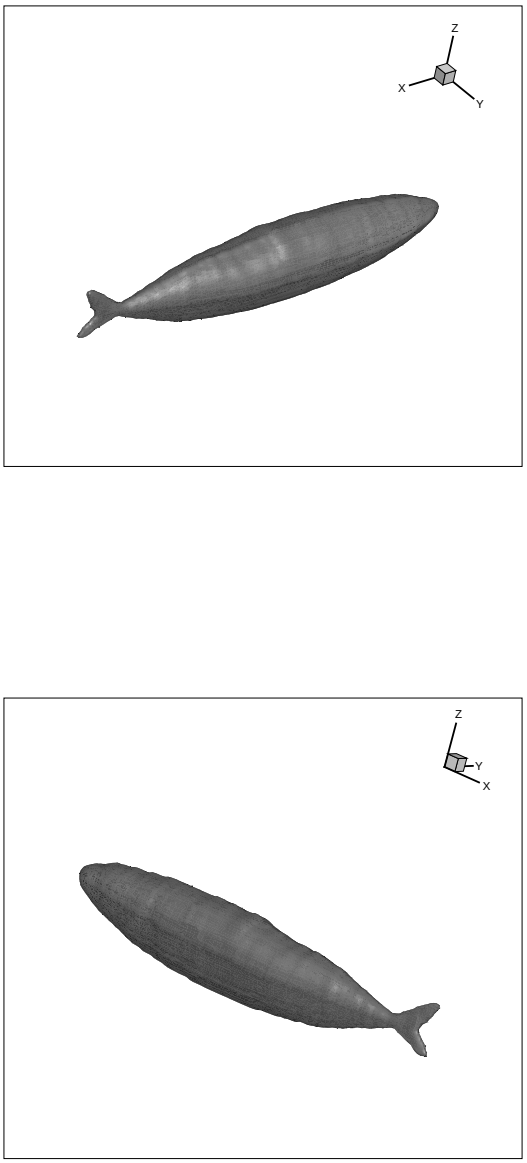


FIGURE 5.20 – Reconstruction 3D du maquereau non déformé.

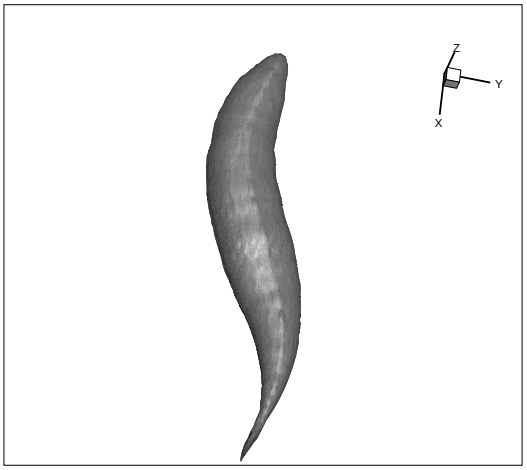


FIGURE 5.21 – Reconstruction 3D du maquereau déformé.

5.4 Conclusion

Au cours de ce chapitre nous avons tout d'abord décrit l'outil mathématique du Squelette. Plusieurs approches peutvent être abordées pour définir ce dernier. Nous avons introduit un aperçu de trois méthodes : la méthode des amincissements successifs, la méthode de type feu de prairie et la méthode basée sur la fonction distance. Étant donné que nous travaillons avec la méthode Level Set, cette dernière approche du squelette fut adoptée.

Nous avons ensuite présenté brièvement le logiciel développé par Cédric Galusinski et Christian Nguyen, logiciel permettant à partir d'une image d'obtenir sa fonction Level Set et son squelette associés, que nous allons utiliser pour reconstruire nos géométries tridimensionnelles de poissons.

En effet, afin de simuler la nage du poisson en 3D, nous avons vu que approchant la coupe transversale d'un poisson par une ellipse, l'utilisation du logiciel sur les vues de dessus et de profil du poisson nous permet de reconstituer une forme approchée du nageur. Cette reconstitution se base sur trois hypothèses fondamentales et, grâce à l'introduction de l'outil du squelette médian, elle nous permet de suivre lagrangiennement le poisson et par conséquent de lui attribuer une loi de nage. Nous avons illustré ce cas test en l'appliquant à la construction et nage d'un chinchard.

Enfin, souhaitant prolonger la reconstruction vers une géométrie complètement réaliste de poisson, nous avons décidé de ne plus utiliser l'approximation d'ellipse pour la vue de face mais bien de l'utiliser telle qu'elle est. Ainsi, à partir des trois images de poisson (face, profil et dessus), nous avons vu comment nous avons reconstruit un poisson en 3D. Cette reconstruction utilise un grand nombre de données issues de la fonction Level Set pour recréer l'ensemble des coupes transversales du corps du poisson, et des homothéties et translations sur la coupe transversale de référence. Nous avons enfin appliqué cette reconstruction au cas d'un maquereau commun. Notons pour terminer que cette étude

constitue un point de départ à de futures perspectives. L'utilisateur intéressé pourra utiliser cette reconstruction pour la nage réaliste d'un poisson grâce à l'outil du *transport optimal* qui lui permettra de connaître la vitesse de déformation du poisson.

Conclusion générale et Perspectives

Durant cette thèse, nous nous sommes intéressés à la modélisation d'interactions fluide-structure dans le cadre des fluides incompressibles et d'une structure *a priori* quelconque. Nous avons mis en place un schéma numérique couplant les équations fluides et solides que nous avons par la suite appliqué à la simulation bi-et-tri-dimensionnelle d'un nageur.

Après avoir introduit quelques notions utiles, nous avons dans un premier temps effectué un état de l'art des différentes méthodes de modélisation des interactions fluide-structure. Nous nous sommes donc tout d'abord intéressés aux méthodes de type frontières immergées. Parmi elles, nous avons détaillé la méthode originelle présentée par Peskin, la pénalisation et les méthodes Ghost-Cell et Cut-Cell. La description de ces approches nous a permis de constater leur grande variété. Nous avons également présenté les méthodes ALE et de maillages adaptatifs dans le but d'avoir un large aperçu des approches possibles à la disposition de tout numéricien. Toutes ces méthodes ont leur *pro and cons* et il revient à l'utilisateur de choisir celle qu'il est préférable d'aborder suivant le contexte physique, les objectifs qu'il s'est fixé et les contraintes d'implémentation qu'il est prêt à surmonter.

Nous avons ensuite présenté un méthode de type frontières immergées que nous avons développée. Après avoir détaillé la résolution numérique des équations de Navier Stokes, nous avons décrit notre nouvelle approche : *la méthode IPC*. Elle s'inspire des méthodes de pénalisation et Ghost-Cell. Nous avons cherché à conserver la simplicité d'implémentation de la méthode de pénalisation tout en bénéficiant de la précision de l'approche Ghost-Cell. En effet, souhaitant modéliser des écoulements fluides en présence de structures mobiles et déformables, en 3D, nous nous sommes naturellement orientés vers un approche de type frontières immergées afin de travailler sur des maillages cartésiens et d'utiliser des schémas numériques simples, facilement parallélisables par la suite.

La méthode IPC se base sur le principe d'une correction à apporter sur la vitesse des ghost cells afin que celles-ci conservent le gradient de vitesse à l'interface.

Nous avons pu constater que la méthode IPC est du second ordre en espace sur la vitesse, globalement, et localement. Dès lors, nous l'avons validée en comparant ses résultats à ceux de la littérature sur de multiples cas tests canoniques. Nous avons ainsi modélisé des cas tests faisant intervenir un ou deux corps rigide(s) fixe(s), un corps impulsivement mis en mouvement. Nous avons comparé ce résultat à la pénalisation classique et avons pu constater que nous obtenions un profil de trainée plus stable, car présentant moins d'oscillations. Enfin, nous avons simulé la chute d'une bille dans une cavité sous l'effet de la gravité. Dans l'ensemble, les résultats obtenus sont en bon accord avec la littérature.

En conclusion, la méthode IPC est simple d'implémentation et du second ordre en vitesse. Néanmoins, l'ordre en pression reste faible localement. Un travail reste donc à effectuer ici afin d'augmenter l'ordre en pression et diminuer son erreur locale. Une approche pourraît être la résolution des équations de Navier Stokes incompressibles sans

pénalisation, avec l'imposition précise des conditions de bord de pression au niveau de l'interface (conditions qui sont pour le moment gérées grâce à la pénalisation). Ainsi, les corrections de vitesse et de pression seraient exactes. Enfin, on appliqueraient la correction de la vitesse des ghost cells afin d'avoir la bonne vitesse à l'interface lors des calculs de gradient. De plus, il serait intéressant dans une future étude d'évaluer l'ordre du schéma dans le cas d'un corps en mouvement et de comparer l'erreur obtenue à une pénalisation classique ou au cas du corps fixe. Enfin, une validation de l'ordre pour des études 3D serait également souhaitable.

Dans la seconde partie de cette thèse, nous avons choisi de nous intéresser à la simulation de nageurs de type poisson.

Dans un premier temps, nous nous sommes placés en 2D. Nous avons cherché à coupler la simulation de la nage du poisson dans un fluide avec des algorithmes mathématiques d'optimisation. Par conséquent, après avoir présenté les différents types de nage et d'optimisation, nous nous sommes placés dans le cadre des poissons à nage ondulatoire de type BCF. Nous avons couplé la modélisation de la nage avec des algorithmes d'optimisation non déterministes (simplexe et recuit-simulé) afin d'optimiser la forme et/ou la nage du poisson pour la minimisation de la puissance moyenne dépensée sur une distance donnée, de l'énergie dépensée sur cette même distance, ou encore la maximisation de l'index propulsif sur toute la nage ou en ne considérant que le régime établi. De tels calculs peuvent être particulièrement intéressants dans le domaine de la robotique sous-marine afin de choisir une forme et/ou une loi de nage du poisson qui consommerait un minimum de carburant par exemple. Lorsque nous avons considéré l'optimisation de deux paramètres de nage, nous avons observé l'apparition de direction de convergence pour les différents éléments de l'optimisation, pour les deux méthodes d'optimisation considérée. Nous nous sommes ensuite concentrés sur la méthode du simplexe car celle-ci est moins couteuse en temps de calculs. Les directions de convergence obtenues par la méthode du simplexe sont restées semblables pour différentes données initiales et paramètres de simplexe. Concernant l'optimisation de trois paramètres, nous nous sommes concentrés sur la méthode simplexe. En choisissant judicieusement notre simplexe initial, nous avons constaté que nous parvenions une fois encore à optimiser les valeurs de la fonctionnelle, et que nous retrouvions les directions de convergence obtenues lors de l'optimisation à deux paramètres et des valeurs cohérentes avec la réalité. Par conséquent, nous sommes parvenus à coupler un algorithme d'optimisation mathématique avec un solveur de modélisation d'interactions fluide-structure. Nous nous sommes néanmoins limités à trois paramètres de contrôle. Il serait donc intéressant de parvenir à ajuster ces paramètres des algorithmes afin d'optimiser l'ensemble des paramètres, et de comparer les résultats obtenus à ceux utilisant une méthode déterministe.

Enfin, nous nous sommes intéressés à la simulation 3D de la nage d'un poisson. Cette étude, effectuée en collaboration avec l'IMATH de Toulon, utilise l'outil mathématique du squelette. En conséquence, après avoir introduit différentes méthodes de détermination du squelette, nous avons choisi de reconstruire un poisson à partir de deux photos de ce

dernier : une photo de profil et une de dessus. Puis, en approchant la vue de face par une ellipse, nous avons été en mesure de construire un maillage lagrangien 3D propre au poisson. Grâce à ce maillage lagrangien, nous avons pu déformer le poisson ainsi obtenu afin de lui appliquer une loi de nage. Nous avons illustré cette étude dans le cas d'un chinchard dont nous avons simulé la nage. Ce travail nous a permis de visualiser les différences entre les résultats 2D et 3D, en particulier l'écartement des tourbillons créés par le poisson dans son sillage lors des simulations 3D, qui restent très compacts en 2D.

Il serait intéressant par la suite de coupler cette modélisation tridimensionnelle aux algorithmes d'optimisation utilisés dans l'étude 2D afin d'observer les profils obtenus et de les comparer au cas 2D. De plus, l'optimisation 3D serait davantage réaliste.

C'est dans ce soucis de réalisme que nous avons développé une dernière étude. Nous avons étudié la reconstruction complète du poisson à partir des trois photos, autrement dit, en n'approchant plus la vue de face par une ellipse mais en considérant la coupe transversale réelle d'un poisson. Nous avons alors été en mesure de reconstruire complètement la géométrie du nageur, ce que nous avons illustré avec le cas d'un maquereau. Cette reconstruction constitue un outil pour une simulation de nage réaliste. Enfin, il serait intéressant d'ajouter la reconstruction de nageoires dans la géométrie afin d'avoir la possibilité de simuler la nage d'une plus grande variété de poissons.

Bibliographie

[1] Anderson, D., McFadden, G., and Wheeler, A. Diffuse interface methods in fluid mechanics. *Annu. Rev. Fluid Mechanics 30* (1998), 139–165. (Cité en page 20.)

[2] Angot, P. *Contribution à l'étude des transferts thermiques dans des systèmes complexes aux composants électroniques.* PhD thesis, Université de Bordeaux I, 1989. (Cité en page 24.)

[3] Angot, P., Bruneau, C.-H., and Fabrie, P. A penalization method to take into account obstacles in incompressible flows. *Numer. Math. 81*, 4 (1999), 497–520. (Cité en pages 25 et 62.)

[4] Arquis, E., and Caltagirone, J. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieux poreux : application la convection naturelle. *C.R. Acad. Sci. Paris II 299* (1984), 1–4. (Cité en page 24.)

[5] Askes, H., and Rodriguez-Ferran, A. A combined rh-adaptive scheme based on domain subdivision. formulation and linear examples. *Int. J. Numer. Methods Eng. 51* (2001), 253–273. (Cité en page 48.)

[6] Askes, H., Sluys, L., and de Jong, B. Remeshing techniques for r-adaptive and combined h/r-adaptive analysis with application to 2d/3d crack propagation. *Struct. Eng. Mech. 12* (2001), 475–490. (Cité en page 48.)

[7] BAINBRIDGE, R. The speed of swimming of fish as related to size and to the frequency and amplitude of the tail beat. *J. Exp. Biol 35* (1958), 109–133. (Cité en page 104.)

[8] Balaras, E. Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Computers and Fluids 33* (2004), 375–404. (Cité en page 26.)

[9] Belytschko, T., Kennedy, J., and Schoeberle, D. Quasi-eulerian finite element formulation for fluid-structure interaction. *J. Press. Vessel Technol.-Trans. ASME* (1980), 62–69. (Cité en page 46.)

[10] Berger, M. *Adaptive mesh refinement for hyperbolic partial differential equations.* PhD thesis, Stanford University, 1982. (Cité en page 49.)

[11] Berger, M., and Aftosmis, M. Aspects (and aspect ratios) of cartesian mesh methods, 1998. (Cité en page 20.)

[12] Bergmann, M. *Optimisation aérodynamique par réduction de modèle POD et contrÃ´le optimal. Application au sillage laminaire d'un cylindre circulaire.* PhD thesis, Institut National Polytechnique de Lorraine, 2004. (Cité en pages 8, 70 et 100.)

[13] Bergmann, M., and Iollo, A. Modeling and simulation of fish-like swimming. *J. Comput. Phys. 230* (2011), 329–348. (Cité en pages ix, 91, 92, 93, 94 et 109.)

[14] Bernhardt, A., Pihuit, A., Cani, M., and Barthe, L. *Matisse : Painting 2D regions for modeling free-form shapes.* 2008. (Cité en page 121.)

[15] Beyer, R., and Leveque, R. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J. Numer. Anal. 29* (1992), 332–364. (Cité en page 23.)

[16] Blum, H. *Models for the perception of speech and visual forms.* MIT Press, 1967. (Cité en pages 120 et 122.)

[17] Bouchon, F., Dubois, T., and James, N. A second order cut-cell method for the numerical simulation of 2d flows past-obstacles. *Computers and Fluids* (2012). (Cité en pages xi, 7, 37, 38, 39 et 41.)

[18] Boumediene, F. *Méthode asymptotique numérique et technique de réduction de modèles pour les vibrations non linéaires de plaques minces amorties.* PhD thesis, Université de Bretagne Sud, 2010. (Cité en page 8.)

[19] Braza, M., Chassaing, P., and Minh, H. H. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journ. Fluid. Mech. 165* (1986). (Cité en page 70.)

[20] Breder, C. *The locomotion of fishes.* Zoologica, vol 4, p159-256, 1926. (Cité en page 86.)

[21] Calabi, L., and Hartnett, W. Shape recognition, prairie fires, convex deficiencies and skeletons. *Am. Math. Monthly 75* (1968), 335–342. (Cité en page 120.)

[22] Castro-Diaz, M., Bourouchaki, H., George, P., Hecht, F., and Mohammadi, B. Anisotropic adaptative mesh generation in two dimensions for cfd. *Proceedings of the Third ECCOMAS Computational Fluid Dynamics Conference, Paris, 9â13 September* (1996), 181–186. (Cité en page 48.)

[23] Chabannes, V., calo Pena, G., and Prud'homme, C. High order fluid-structure interaction in 2d and 3d. application to blood flow in arteries. *Rapport de recherche hal 00657622* (2012). (Cité en page 49.)

[24] Chantalat, F., Bruneau, C.-H., Galusinski, C., and Iollo, A. Level-set and adjoint-based optimization methods for inverse problems. *in : 6th International Congress on Industrial and Applied Mathematics, 16-20 July, Zurich* (2007). (Cité en page 7.)

[25] Cheny, Y., and Botella, O. The ls-stag method : a new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. *Journal of Computational Physics, 229* (2010), 1043–1076. (Cité en pages 7, 36 et 38.)

[26] Cheny, Y., and Botella, O. On the treatment of complex geometries in a cartesian grig flow solver with the level set method. *European Conference on Computational Fluid Dynamics ECCCOMAS CFD 06,* (206). (Cité en pages 36 et 38.)

[27] Choi, J., Oberoi, R., Edwards, J., and Rosati, J. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics, 224* (2007), 757–784. (Cité en page 26.)

[28] Chorin, A. Numerical solution of the navier stokes equations. *Math. Comput. 22* (1968), 746–762. (Cité en pages 27 et 55.)

[29] Clarke, D., Salas, M., and Hassan, H. Euler calculations of muti-element airfoils using cartesian grids. *AIAA J. 24* (1986), 1128–1135. (Cité en pages 20 et 33.)

[30] Coquerelle, M., and Cottet, G.-H. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys. 227*, 21 (Nov. 2008), 9121–9137. (Cité en pages ix, xi, 41, 78, 79, 80 et 81.)

[31] Cottet, G., and Koumoutsakos, P. *Vortex Methods*. Cambridge University Press, 2000. (Cité en page 12.)

[32] Dehkordi, D., Moghaddam, H., and Jafari, H. Numerical simualtion of flow over two circular cylinders in tandem arrangement. *Journal of Hydrodynamics 23* (2011), 114–126. (Cité en pages 77 et 78.)

[33] DeZeeuw, D., and Powell, K. An adaptative cartesian mesh for the euler equations. *Journal of Computational Physics 245* (1992). (Cité en page 50.)

[34] Ding, H., Shu, C., and Yeo, K. Numerical simulation of flows around two circular cylonders by mesh-free least square-based finite difference methods. *Int. J.for Num. Meth. in Fluids 53* (2007), 305–332. (Cité en page 77.)

[35] Donea, J., Huerta, A., Ponthot, J., and Rodriguez-Ferran, A. *Encyclopedia of Computational Mechanics*. Wiley publishers, 2004. (Cité en pages viii, 42, 43, 44 et 47.)

[36] Eude, Y. *Développement d'un outil de simulation numérique des écoulements r''eactifs sur maillage auto-adaptatif et son application à un moteur à détonation continue*. PhD thesis, Université d'Orléans, 2011. (Cité en pages viii, 50 et 51.)

[37] Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. Combined immersed boundary finite difference methods for three dimensional complex flow simulations. *J. Comput. Phys. 161* (2000), 35–60. (Cité en pages vii et 26.)

[38] Ferziger, J., and Perié, M. *Computational Methods for Fluid Dynamics*. Third Ed. Springer Verlag, Berlin, Heidelberg, 2001. (Cité en page 27.)

[39] Fortin, M., Vallet, M., Dompierre, J., Bourgault, Y., and Habashi, W. Anisotropic mesh adaptation : theory, validation and applications. *Proceedings of the Third ECCOMAS Computational Fluid Dynamics Conference, Paris, 9â13 September* (1996), 174–180. (Cité en page 48.)

[40] Franck, R., and Lazarus, R. *Mixed Eulerian-Lagrangian method*. Methods in Computational Physics, Alder B, Fernbach S and Rotenberg M (eds), vol. 3. Academic Press : New York,47-67, 1964. (Cité en page 42.)

[41] Galusinski, C., and Nguyen, C. Skeleton and level set for channel reconstruction and flow simulation. *Graphical Model*, soumis. (Cité en pages x, 122, 123 et 124.)

[42] Gasteiger, R., Neugebauer, M., Kubish, C., and Preim, B. Adapted surface visualization of cerebral aneurysms with embedded blood flow information. *Eurographics Workshop on visual computing for biology and medecine* (2010), 25–32. (Cité en page 121.)

[43] Ghias, R., Mittal, R., and Dong, H. A sharp interface immersed boundary method for compressible viscous flows. *J. Comput. Phys. 225* (2007), 528–553. (Cité en pages 7 et 33.)

[44] Gibou, F., Fedkiw, R., Cheng, L., and Kang, M. A second order accurate symmetric discretization of the poisson equation on irregular domains. *J. Comput. Phys. 176* (2002), 205–227. (Cité en page 29.)

[45] Gilmanov, A., Sotiropoulos, F., and Balaras., E. A general reconstruction algorithm for simulating flows with complex 3d immersed boundaries on cartesian grids. *J. Comput. Phys. 191* (2003), 660–669. (Cité en page 26.)

[46] Gray, J. Studies in animal locomotion. vi. the propulsive power of the dolphin. *J. Exp. Biol. 13* (1936), 192–199. (Cité en page 2.)

[47] Haber, R. A mixed eulerian-lagrangian displacement model for large-deformation analysis in solid mechanics. *Comput. Methods Appl. Mech. Eng. 43(3)* (1984), 277–292. (Cité en page 47.)

[48] Haber, R., and Hariandja, B. Computational strategies for nonlinear and fracture mechanics problems. an eulerian-lagrangian finite element approach to large deformation frictional contact. *Comput. Struct. 20* (1985), 193–201. (Cité en page 47.)

[49] Hall, R. Fast parallel thinning algorithms : parallel speed and connectivity preservation. *Commun. ACM 32* (1989), 124–131. (Cité en page 121.)

[50] Henderson, R. Details of the drag curve near the onset of vortex shedding. *Phys. Fluids 7* (1995), 2102–2104. (Cité en page 70.)

[51] Hirt, C., Amsden, A., and Cook, J. An arbitrary lagrangian- eulerian computing method for all flow speeds. *J. Comput. Phys 14* (1974), 227–253, Réimprimé dans *J. Comput. Phys. 135* (1997) :203 –216. (Cité en page 42.)

[52] Hisada, M., Belyaev, A., and Kunii, T. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum 21* (2002), 689–700. (Cité en page 120.)

[53] Hoffmann, C. *Geometric and solid modeling.* Morgan-Kaufmann, San Mateo, CA, 1989. (Cité en page 120.)

[54] Huerta, A., and Liu, W. Viscous flow structure interaction. *J. Press. Vessel Technol.-Trans. ASME 110(1)* (1988), 15–21. (Cité en page 47.)

[55] Huerta, A., Rodriguez-Ferran, A., Dìez, P., and Sarrate, J. Adaptive finite element strategies based on error assessment. *Int. J. Numer. Methods Eng. 46* (1999), 1803–1818. (Cité en page 48.)

[56] HUNTERA, J., and ZWEIFEL, J. Swimming speed, tail beat frequency, tail beat amplitude, and size in jack mackerel, truchurus symmetricus, and other fishes. *FISHERY BULLETIN 69* (1971), 253–266. (Cité en page 104.)

[57] Iaccarino, G., and Verzicco, R. Immersed boundary technique for turbulent flow simulations. *Appl. Mech. Rev. 56* (2003), 331–347. (Cité en page 31.)

[58] Kalitzin, G., Iaccarino, G., and Khalighi, B. Towards an immersed boundary solver for rans simulations. *AIAA Pap. 2003-0770* (2003). (Cité en page 33.)

[59] Kim, J., and Moin, P. Application of a fractional step method to incompressible navier stokes equations. *J. Comput. Phys. 59* (1985), 308–329. (Cité en page 27.)

[60] Koumoutsakos, P. *Direct Numerical Simulations of Unsteady separated Flows using Vortex Methods*. PhD thesis, California Institute of Technology, 1993. (Cité en page 12.)

[61] Koumoutsakos, P., and Leonard, A. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech. 296* (1995), 1–38. (Cité en pages viii, ix, 70, 71, 73, 74, 76 et 81.)

[62] Lagarias, J., Reeds, J., Wright, M., and Wright, P. Convergence properties of nelder-mead simplex method in low dimensions. *Siam J. Optim 9* (1998), 112–147. (Cité en page 102.)

[63] Lai, M., and Peskin, C. An immersed boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Physics 171* (2001), 705–719. (Cité en pages xi, 23 et 41.)

[64] Leveque, R., and Li, Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Numerical Analysis 31*, 4 (1994), 1019–1044. (Cité en pages 20 et 40.)

[65] Liu, W., and Chang, H. Efficient computational procedures for long-time duration fluid-structure interaction problems. *J. Press. Vessel Technol.-Trans. ASME 106* (1984), 317–322. (Cité en page 47.)

[66] Liu, W., and Chang, H. A method of computation for fluid structure interaction. *Comput. Struct. 20* (1985), 311–320. (Cité en page 49.)

[67] Mahir, N., and Altac, Z. Numerical investigation of convective heat transfer in unsteady flow past two cylinders in tandem arrangements. *Int. J. of Heat and Fluid Flow 29* (2008), 1309–1318. (Cité en pages 77 et 78.)

[68] Meneghini, J., and Satara, F. Numerical simulation of flow interference between two cylinders in tandem and side-by-side arrangements. *J.of Fluids and Structures 15* (2001), 327–350. (Cité en pages 77 et 78.)

[69] Milcent, T. *Une approche eulérienne du couplage fluide-structure, analyse mathématique et application en biomécanique*. PhD thesis, Université Joseph Fourier, 2009. (Cité en pages vii, 11 et 13.)

[70] Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and von Loebbecke, A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys. 227* (2008), 4825–4852. (Cité en pages vii, xi, 7, 31, 32, 41, 54, 63 et 70.)

[71] Mittal, R., and Iaccarino, G. Immersed boundary methods. *Annu. Rev. Fluid. Mech* (2005), 1–27. (Cité en pages vii et 24.)

[72] Mittal, R., Seshadri, V., and Udaykumar, H. Flutter, tumble and vortex induced autorotation. *Theor. Comput. Fluid Dyn. 17* (2004), 165–170. (Cité en pages 33 et 36.)

[73] Mittal, S., Kumar, V., and Raghuvanshi, A. Unsteady incompressible flows past two cylinders in tandem and staggered arrangements. *Int. J. for Num. Meth. in Fluides 25* (1997), 1315–1344. (Cité en page 77.)

[74] Mohd-Yusof, J. Combined immersed boundaries/b-splines methods for simulations of flows in complex geometries. *CTR Annual Research Briefs, NASA Ames/Stanford University* (1997). (Cité en page 26.)

[75] Moudid, L. A. *Couplage fluide-structure pour la simulation nuérique des écoulements fluides dans une conduite à parois rigides ou élastiques, en présence d'obstacles ou non.* PhD thesis, California Institute of Technology, 2008. (Cité en page 49.)

[76] Müller, J. Anisotropic adaptation and multigrid for hybrid grids. *Int. J. Numer. Methods Fluids 40* (2002), 445–455. (Cité en page 48.)

[77] Nocebal, J., and Wright, S. *Numerical Optimization.* Springer Series in Operations research, 1999. (Cité en page 101.)

[78] Noh, W. *A time-dependent two-space dimensional coupled Eulerian-Lagrangian code.* Methods in Computational Physics, Alder B, Fernbach S and Rotenberg M (eds), vol. 3. Academic Press : New York,117â179, 1964. (Cité en page 42.)

[79] Nomura, T., and Hugues, T. An arbitrary lagrangian-eulerian finite element method for interaction of fluid and a rigid body. *Comput. Methods Appl. Mech. Eng. 95* (1992), 115–138. (Cité en page 49.)

[80] Osher, S., and Sethian, J. A. Fronts propagating with curvature-dependent speed : Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys. 79*, 12 (1988). (Cité en page 13.)

[81] P. Ploumhans, G. W. Vortex methods for high-resolution simulations of viscous flow past bluff bodies in general geometry. *Journal of Computational Physics 165* (2000), 354–406. (Cité en pages 12 et 68.)

[82] Peskin, C. Flow patterns around the heart valves. *Journal of Computational Physics 10* (1972), 252–271. (Cité en pages 3 et 20.)

[83] Peskin, C. The fluid dynamics of heart valves : experimental, theoretical and computational methods. *Annu. Rev. Fluid Mech. 14* (1981), 235–259. (Cité en page 23.)

[84] Peskin, C. The immersed boundary method. *Acta Numer. 11* (2002), 1–39. (Cité en pages 22 et 23.)

[85] Peskin, C., and Printz, B. Improved volume conservation in the computation of flows with immersed elastic boundaries. *J. Comput. Phys. 105* (1993), 33–46. (Cité en page 24.)

[86] P.Fedkiw, R., Aslam, T., Merriman, B., and Osher, S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys. 152* (1999), 457–492. (Cité en page 33.)

[87] Poncet, P. *Méthodes particulaires pour la simulation des sillages tridimensionnels.* PhD thesis, Université Joseph Fourier, 2001. (Cité en page 12.)

[88] Rodriguez-Ferran, A., Pérez-Foguet, A., and Huerta, A. Arbitrary lagrangian-eulerian (ale) formulation for hyperelastoplasticity. *Int. J. Numer. Methods Eng. 53(8)* (2002), 1831–1851. (Cité en pages 47 et 48.)

[89] Saad, Y., and Schultz, M. Gmres : a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput. 7* (1986), 856–869. (Cité en page 55.)

[90] Saiki, E., and Biringen, S. Numerical simulation of a cylinder in uniform flow : application of a virtual boundary method. *J. Comput. Phys. 123* (1996), 450–465. (Cité en page 23.)

[91] Sarrate, J., Huerta, A., and Donea, J. Lagrangian-eulerian formulation for fluid rigid-body interaction. *Comput. Methods Appl. Mech. Eng. 190* (2001), 3171–3188. (Cité en page 49.)

[92] Scardovelli, R., and Zaleski, S. Direct numerical simulation of free-surface and interfacial flows. *Annu. Rev. Fluid Mechanics 31* (1999), 567–603. (Cité en page 20.)

[93] Schumann, C., Neugebauer, M., Bade, R., Preim, B., and Peitgen, H. Implicit vessel surface reconstruction for visualization and simulation. *Internation Journal of Computer Assisted Radiologu and Surgery 2* (2008), 275–286. (Cité en page 121.)

[94] Sfakiotakis, M., Lane, D., and Davies, J. B. C. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering 24*, 2 (1999), 237–252. (Cité en pages ix, 87, 88, 89 et 94.)

[95] Sharman, B., Lien, F., and Davidson, L. Numerical predictions of low reynolds number flows over two tandem circular cylinders. *Int. J. for Num. Meth. and Fluids 47*, 5 (2005), 423–447. (Cité en page 77.)

[96] Slaouti, A., and Stansby, P. Flow around two circular cylinders by random-vortex method. *J. of Fluids and Structures 6*, 6 (1992), 641–670. (Cité en pages 77 et 78.)

[97] Sonka, M., Hlavac, V., and Boyle, R. *Image processing, analysis and machine vision.* PWS Publishing, Pacific Grove, seconde édition, 1999. (Cité en page 120.)

[98] Sussman, M., Smereka, P., and Osher, S. A level set approach for computing solutions to incompressible 2-phade flow. *Journal of Computational Physics 114* (1994), 146–159. (Cité en page 20.)

[99] Sweet, R. A cyclic reduction algorithm for solving block tridiagonal systems on arbitrary dimensions. *SIAM J. on Numer. Ana. 14* (1977), 706–720. (Cité en page 56.)

[100] Tau, E. a second order projection method for incompressible navier stokes equations in arbitrary domains. *Journal of Computational Physics 115* (1994), 147–152. (Cité en page 36.)

[101] Temam, R. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires. *Arch. Rational Mech. Anal. 32* (1969), 135–153. (Cité en pages 27 et 55.)

[102] Tozaki, T., Kawata, Y., Niki, N., Ohmatsu, H., and Moriyama, N. 3d visualization of blood vessels and tumor using thin slice ct. *IEEE Nuclear Science Symposium and Medical Imaging Conference 3* (1995), 1470–1474. (Cité en page 121.)

[103] Trulio, J. Theory and structure of the afton codes. *Report AFWL-TR-66-19, Air Force Weapons Laboratory : Kirtland Air Force Base* (1966). (Cité en page 42.)

[104] Tryggvason, G., Bunner, B., Esmaeeli, A., and Al-Rawahi, N. Computational of multiphase flows. *Adv. Appl. Mech. 39* (2003), 91 –120. (Cité en page 20.)

[105] Tseng, and Ferziger. A ghost cell immersed boundary method for flow in complex geometry. *J. Comput. Phys. 192* (2003), 593 – 623. (Cité en pages vii, 27, 30 et 31.)

[106] Tucker, P., and Pan, Z. A cartesian cut cell method for incompressible viscous flow. *Appl. MAth. Modell. 24* (2000), 591–606. (Cité en page 36.)

[107] Udaykumar, H., Kan, H., Shyy, W., and Tran-Son-Tay, R. Multiphase dynamics in arbitrary geometries on fixed cartesian grids. *Journal of Computational Physics 137* (1997), 366–405. (Cité en page 34.)

[108] Udaykumar, H., Mittal, R., and Rampunggoon, P. Interface tracking finite volume method for complex solid- fluid interactions on fixed meshes. *Commun. Numer. Meth. Engng 18* (2002), 89–97. (Cité en page 36.)

[109] Udaykumar, H., Mittal, R., Rampunggoon, P., and Khanna, A. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics 174* (2001), 345–380. (Cité en pages 33, 34 et 63.)

[110] Udaykumar, H., Mittal, R., and Shyy, W. Computational of solid-liquid phase fronts in the sharp interface limit on fixed grids. *Journal of Computational Physics 153* (1999), 535–574. (Cité en pages 34 et 63.)

[111] Udaykumar, H., Shyy, W., and Rao, M. M. Elafint : a mixed eulerian-lagrangian method for fluid flows with complex and moving boundaries. *Int J. Numer Methods Fluids 22* (1996), 691–705. (Cité en pages 20 et 33.)

[112] Unversi, S., and Truggvason, G. A front tracking method for viscous incompressible multifluid flows. *Journal of Computational Physics 100* (1992), 25–42. (Cité en page 23.)

[113] van Haaren, M., Stoker, H., van den Boogaard, A., and Huétink, J. The ale method with triangular elements : Direct convection of integration point values. *Int. J. Numer. Methods Eng. 49(5)* (2000), 697–720. (Cité en page 47.)

[114] Verzicco, R., Iaccarino, G., Fatica, M., , and Orlandi, P. Flow in an impeller stirred tank using an immersed boundary method. *Center for Turbulence Research Annual Research Briefs* (2000), 251–261. (Cité en page 26.)

[115] Vigneaux, P. *Méthodes Level Set pour les problèmes d'interfaces en microfluidique.* PhD thesis, Université Bordeaux I, 2007. (Cité en pages vii, 7, 15 et 16.)

[116] Webb, P., Maddock, L., Bone, Q., and Rayner, J. *Mechanics and physiology of animal swimming.* Cambridge, U.K., Cambridge Univ., 1994. (Cité en page 86.)

[117] Weynans, L. *Méthode particulaire multi-niveaux pour la dynamique des gaz, application au calcul d'écoulements multifluides*. PhD thesis, Université Joseph Fourier, 2006. (Cité en page 12.)

[118] Wieselsberger, C. New data on the laws of fluid resistance. *NACA TN 84* (1922). (Cité en page 70.)

[119] Xia, I. Skeletonization via the realisation of the fire front's propagation ans extintion in digital binary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11* (1989), 1076–1086. (Cité en page 122.)

[120] Xu, S., and Wang, Z. J. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *J. Comput. Phys. 216 - 2* (2006), 454–493. (Cité en page 20.)

[121] Yang, Y., and Udaykumar, H. Sharp interface castesian gird method iii : solidification of pure materials and binary solutions. *J. Comput. Phys. 210* (2005), 55–74. (Cité en page 20.)

[122] Ye, T., Mittal, R., Udaykumar, H., and Shyy, W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys. 156* (1999), 209–240. (Cité en pages vii, xi, 20, 33, 34, 35, 41 et 63.)

[123] Zeew, D., and Powell, K. An adaptively-refined cartesian mesh solver for euler equations. *AIAA Pap.* (1991). (Cité en page 20.)

[124] Zhang, N., and Zheng, Z. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics 221* (2007), 250–268. (Cité en page 26.)

[125] Zhang, Q., and Hisada, T. Analysis of fluid-structure interaction problems with structural buckling and large domain changes by ale finite element method. *Comput. Methods Appl. Mech. Eng. 190* (2001), 6341–6357. (Cité en page 49.)

[126] Zuzio, D. *Direct numerical simulation of two phase flows with adaptative mesh refinement*. PhD thesis, Université de Toulouse, 2010. (Cité en pages viii et 51.)