



HAL
open science

CONTRIBUTIONS A LA SIMULATION NUMERIQUE EN MECANIQUE DES FLUIDES ET A LA REDUCTION DE MODELE

Michel Bergmann

► **To cite this version:**

Michel Bergmann. CONTRIBUTIONS A LA SIMULATION NUMERIQUE EN MECANIQUE DES FLUIDES ET A LA REDUCTION DE MODELE. Mécanique des fluides [physics.class-ph]. université de Bordeaux, 2014. tel-01113058

HAL Id: tel-01113058

<https://inria.hal.science/tel-01113058v1>

Submitted on 4 Feb 2015

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

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

HABILITATION A DIRIGER DES RECHERCHES

Université de Bordeaux

Spécialité : Mathématiques appliquées et Calcul Scientifique

CONTRIBUTIONS A LA SIMULATION NUMERIQUE EN MECANIQUE DES FLUIDES ET A LA REDUCTION DE MODELE.

présentée par

Michel BERGMANN

Soutenance prévue le 13 juin 2014 devant le jury composé de :

C.-H. BRUNEAU	Professeur, Université de Bordeaux	Examineur
T. COLIN	Professeur, Institut Polytechnique de Bordeaux	Examineur
G.-H. COTTET	Professeur, Université de Grenoble	Examineur
F. GIBOU	Professeur, University of California at Santa Barbara, USA	Rapporteur
A. HAMDOUNI	Professeur, Université de La Rochelle	Examineur
A. IOLLO	Professeur, Université de Bordeaux	Examineur
P. SAGAUT	Professeur, Université Pierre et Marie CURIE	Rapporteur
M. TUCSNAK	Professeur, Université de Lorraine	Rapporteur

Table des matières

I	Réduction de modèle et bases réduites	5
1	Contributions aux modèles réduits basés sur la Décomposition Orthogonale aux valeurs Propres	9
1.1	Du modèle classique vers le modèle à pression	9
1.1.1	Configuration d'étude	9
1.1.2	Modèle réduit standard basé sur la décomposition orthogonale aux valeurs propres . .	10
1.1.3	Un modèle POD avec pression	12
1.2	Amélioration de la stabilité du modèle réduit	17
1.2.1	Synthèse des méthodes de stabilisation existantes	17
1.2.2	Méthodes de stabilisation basées sur le résidu : Modèle $B^{[N_r;K]}$	19
1.2.3	Méthodes SUPG et VMS : modèles $C^{[N_r]}$ et $D^{[N_r]}$	19
1.2.4	Résultats numériques des méthodes de stabilisation	20
1.3	Amélioration de la robustesse de la base POD	21
1.3.1	Une méthode basée sur les sous espaces de Krylov	23
1.3.2	Une méthode hybride DNS/POD ROM	26
1.3.3	Construction d'une base POD robuste : Iterative Centroidal Voronoi Tessellation (ICVT)	30
1.4	Conclusions et perspectives	34
II	Modélisation et simulations numériques en mécanique des fluides	37
2	Modélisation et méthodes numériques pour des fluides en présence d'interfaces	41
2.1	Introduction	41
2.2	Suivi et capture d'interface : choix du maillage	42
2.2.1	Approches Lagrangiennes : suivi d'interfaces	42
2.2.2	Approches Eulériennes : capture d'interface	43
2.3	Fonction level set : la fonction distance signée	44
2.3.1	Transport de la fonction level set	45
2.3.2	Redistanciation	47
2.3.3	Conservation de la masse	51
2.4	Modélisation d'écoulements fluides en présence d'interfaces	52
2.4.1	Modèle avec interfaces fluide/fluide/structure	52
2.4.2	Modèle avec interface fluide/structure	54
2.4.3	Modèle avec interface fluide/fluide	54
2.5	Domaine fictif et interface fluide/structure	55
2.5.1	Configuration d'écoulement et modèle de Navier-Stokes incompressible	55
2.5.2	Approche continue	55
2.5.3	Approche discrète	56
2.5.4	Une méthode hybride au second ordre	58
2.5.5	Résolutions numériques des équations	58
2.6	Modélisation et résolution numérique des interfaces fluide/fluide	62
2.6.1	Interface lisse et <i>Continuum Surface Force</i> (CSF)	62
2.6.2	Interface raide et <i>Ghost Fluid Method</i> (GFM)	63
2.7	Interfaces fluide/fluide/structure et ligne triple	64
2.8	Le code de calcul NaSCar (NAvier-Stokes CARTésien)	64

2.8.1	Description	64
2.8.2	Validations	65
2.9	Conclusions	66
3	Simulations d'écoulements autour d'obstacles déformables : application à la nage de poissons	67
3.1	Introduction	67
3.2	Préliminaires et définitions	68
3.2.1	Introduction	68
3.2.2	Configuration d'écoulement	68
3.2.3	Déformation et déplacement de corps	68
3.2.4	Classification de la nage	70
3.2.5	Représentation paramétrique de la nage de poissons	70
3.3	Simulations bidimensionnelles de nageurs	73
3.3.1	Choix de la résolution en espace	73
3.3.2	Mécanisme de la nage en 2D	74
3.3.3	Simulations numériques de nageurs de type BCF	74
3.3.4	Considérations énergétiques de la nage	77
3.4	Simulations tridimensionnelles de nageurs	79
3.4.1	Convergence de grille	80
3.4.2	Manœuvres : proie/prédateur	81
3.4.3	Efficacité en fonction des paramètres de nage	81
3.4.4	Efficacité en fonction de la rigidité de la nageoire caudale	82
3.4.5	Quelques autres nageurs	85
3.4.6	Simulations numériques de la nage d'un poisson à partir d'images	88
3.5	Conclusions et perspectives	95
4	Etude de quelques systèmes de récupération de l'énergie	99
4.1	Introduction	99
4.2	Modélisation et simulations numériques d'éoliennes	100
4.2.1	Description et contexte de l'étude : collaboration avec VALOREM	100
4.2.2	Modèle simplifié et code de calcul VALDAG	101
4.2.3	Simulations tridimensionnelles et limites de l'étude	105
4.3	Récupération de l'énergie de la houle	109
4.3.1	Un modèle de structure élastique	110
4.3.2	Résolution numérique et couplage	114
4.3.3	Modélisation et simulations numériques d'écoulements biffuides	116
4.3.4	Gestion de la ligne triple	117
4.3.5	Vers les serpents de mer	119
4.4	Conclusions et perspectives	120
5	Conclusion générale et perspectives	123
	Bibliographie	125

Table des figures

1.1	Configuration d'écoulement et cliché de la vorticit� pour $Re = 200$	10
1.2	Spectre de valeurs propres.	15
1.3	Repr�sentation de quelques modes POD en iso vorticit� (gauche) et isobares (droite). Les traits pointill�s repr�sentent des valeurs n�gatives (la pression est arbitrairement initialis�e � z�ro).	15
1.4	RIC des modes de fluctuation.	16
1.5	Spectre des valeurs propres.	16
1.6	Evolution temporelle des coefficient de pr�diction sur 40 p�riodes de lâch� tourbillonnaire pour le mod�le � 11 modes.	16
1.7	Comparaison des cycles limites pour les modes projet�s (NS : \diamond) et les modes pr�dits (ROM : —) sur 1000 p�riodes de lâch� tourbillonnaire, avec 11 modes.	17
1.8	Comparaison des cycles limites pour les modes projet�s (NS : \diamond) et les modes pr�dits (ROM : —) sur 1000 p�riodes de lâch� tourbillonnaire, avec 5 modes.	17
1.9	Comparaison des cycles limites pour les modes projet�s (NS : \diamond) et les modes pr�dits (ROM : —) sur 1000 p�riodes de lâch� tourbillonnaire, avec 3 modes.	17
1.10	Comparaison des modes POD originaux (gauche, mod�le $A^{[5]}$) avec les modes de r�siduals (droite, mod�le $B^{[3;2]}$). Les valeurs n�gatives sont repr�sent�es en pointill�s.	21
1.11	Comparaison des cycles limites DNS projet�s (NS : \diamond) et pr�dits (ROM : —) sur 1000 p�riodes pour les mod�les $B^{[5;2]}$, $C^{[5]}$ et $D^{[5]}$ (tous les cycles sont confondus).	21
1.12	Comparaison des cycles limites DNS projet�s (NS : \diamond) et pr�dits (ROM : —) sur 1000 p�riodes pour les mod�les $B^{[3;2]}$, $C^{[3]}$ et $D^{[3]}$ (tous les cycles sont confondus).	21
1.13	Evolution temporelle de la norme L^2 des r�siduals POD-NS avec 5-modes.	22
1.14	Evolution temporelle de la norme L^2 des r�siduals POD-NS avec 3-modes.	22
1.15	Comparaison de quelques modes POD en iso-vorticit� pour $Re_1 = 100$ (gauche) et $Re_2 = 200$ (droite).	23
1.16	Repr�sentation sch�matique de l'adaptation de la base POD avec une m�thode de type Krylov.	24
1.17	Comparaison de quelques modes POD pour $Re_1 = 50$ (—) et $Re_2 = 300$ (---).	25
1.18	Evolution du crit�re de convergence en fonction du nombre d'it�rations pour l'�quation de Burgers 1D.	25
1.19	Evolution de crit�re de convergence en fonction du nombre d'it�rations pour les �quations de Navier-Stokes 2D.	26
1.20	Repr�sentation sch�matique d'une m�thode hybride DNS-POD ROM pour am�liorer la base POD.	26
1.21	Comparaison des valeurs propres entre tenseur exact (�valu� avec \mathbf{U}) et approch� avec $\tilde{\mathbf{U}}^{[1,\dots,N_r]}$ (avec $N_r = 5$ et $N_r = 11$ modes).	28
1.22	Modification des fonctions de base POD avec l'application lin�aire φ . Repr�sentation en lignes de courant.	29
1.23	Evolution de crit�re de convergence en fonction de nombre de p�riode tourbillonnaires.	29
1.24	Repr�sentation sch�matique de la convergence de la m�thode hybride DNS-POD. cible exacte; --- POD ROM; — DNS.	30
1.25	Coefficient de force normale en fonction de l'angle. R�sultats DNS et exp�rimentaux d'AGARD R702.	30
1.26	Nombre de Mach.	31
1.27	Fonctions de base POD pour la pression (gauche) et la vitesse verticale (droite).	31
1.28	Spectre des valeurs propres pour les composantes de vitesse u et v , la pression p et la vitesse du son a	32

1.29	Exemple d'une itération du processus d'enrichissement de Voronoi. Le sous espace S est représenté.	33
1.30	Echantillonnage des paramètres d'entrée.	33
2.1	Représentation schématique de la méthode de marqueurs.	43
2.2	Configurations des différents types d'écoulements étudiés.	54
2.3	Schématisation de la méthode Cut-Cells en 2D par Ye <i>et al.</i> (1999).	57
2.4	Schématisation de la méthode GCM de Ghias <i>et al.</i> (2007)	57
2.5	Grille collocalisée pour (u, v, p) et MAC pour (U, V, p)	59
2.6	Configuration de la sédimentation d'une sphère avec un exemple de champ de vorticit�.	66
3.1	Configuration d'écoulement g�n�rique.	69
3.2	Classification de la nage des poissons de type BCF (Body and Caudal Fin) par Lindsey (1978).	70
3.3	Exemple de transformation de Karman-Trefftz.	71
3.4	Repr�sentation des axes des ellipses axes $y(x)$ et $z(x)$ qui d�finissent le profil 3D.	71
3.5	Configuration des profils de nage et de manoeuvres.	72
3.6	Exemples de profils de nage.	72
3.7	Convergence de la vitesse de nage en fonction du pas d'espace h	74
3.8	Sch�matisation 2D des m�canismes de la nage d'un poisson.	74
3.9	Repr�sentation du sillage g�n�r� par des nageurs de type BCF pour $Re = 10^3$. $-100 \leq \omega_z \leq 100$	75
3.10	Sch�matisation de l'angle de vision.	76
3.11	Sillage et trajectoire du poisson qui manoeuvre pour atteindre automatiquement la cible (disque gris) pour $Re = 10^3$. $-100 \leq \omega_z \leq 100$	77
3.12	Sillages g�n�r�s par 2 poissons F_1 pour $Re = 10^3$ avec $D = 0.5$. Les figures de gauche et droite correspondent � la nage en phase et antiphase respectivement. $-100 \leq \omega_z \leq 100$, $-1.1 \leq u \leq 0.7$	79
3.13	Visualisation du sillage 3D (iso vorticit�) g�n�r� par un poisson � $Re = 10^4$	80
3.14	Coupe 2D du sillage g�n�r� par une nage 3D.	80
3.15	Evolution de la vitesse moyenne $\langle v \rangle$ et de l'efficacit� moyenne $\langle \eta \rangle$ en fonction $1/h$ (avec h le pas d'espace). Les crochets repr�sentent un op�rateur de moyenne.	81
3.16	Exemple de la chasse par un pr�dateur. La proie nage mais n'esquive pas le pr�dateur.	81
3.17	Evolution de l'efficacit� de nage en fonction de l'amplitude et de la fr�quence de battement de la queue.	82
3.18	D�formation du poisson sur une p�riode T (fr�quence $f = 2.0 Hz$). La m�me loi de nage sinuso�dale est appliqu�e sur tout le corps sauf sur la nageoire caudale entre la section noire et la fin de la queue o� un comportement �lastique est simul�. Le corps noir correspond � une d�formation impos�e, le corps bleu correspond � une nageoire rigide et le corps rouge correspond � une nageoire flexible.	83
3.19	Evolution de l'efficacit� η , du travail utile W_{utile} et du travail total W_{total} en fonction du param�tre de flexibilit� α . Les traits pointill�s correspondent � une d�formation impos�e.	84
3.20	Evolution de l'efficacit� η , du travail utile W_{utile} et du travail total W_{total} en fonction du param�tre de flexibilit� α pour $V = 0.4$. Les traits pointill�s correspondent � une d�formation impos�e pour $V = 0.4$	84
3.21	Sillage g�n�r� par le poisson avec $\alpha = 0.1$ pour $Re = 10^3$	84
3.22	Visualisation de la contribution du travail utile sur le corps du nageur pour une d�formation impos�e (gauche), pour une nageoire rigide ($\alpha = 0.02$, milieu), et pour une nageoire flexible ($\alpha = 0.12$, droite) pour $V = 0.4$. Les clich�s sont extraits aux m�mes instants. Couleurs : gris (0, zones inefficaces), noir (-0.05 , zones efficaces).	85
3.23	Profil de poisson pour diff�rents nombres de liaisons de la nageoire caudale N_L . Le cas $N_L = 60$ correspond � $\alpha = 0.12$	85
3.24	Evolution de l'efficacit� η , du travail utile W_{utile} et du travail total W_{total} en fonction du nombre de liaisons N_L pour $V = 0.4$. Les trait en pointill�s correspond � la d�formation impos�e sur tout le profil pour $V = 0.4$	85
3.25	D�formation de la m�duse pendant une p�riode de nage.	86
3.26	Evolution temporelle de la vitesse de la m�duse.	86
3.27	Evolution temporelle de la force g�n�r�e par la m�duse.	86
3.28	Champ de pression g�n�r� par une m�duse ($z = 0$).	87
3.29	Divergence du champ de vitesse g�n�r� par une m�duse ($z = 0$).	87

3.30	Iso valeurs de vorticit� du sillage tridimensionnel g�n�r� par une m�duse.	87
3.31	Coupe bidimensionnelle du sillage g�n�r� par une m�duse.	87
3.32	Profil d'un poisson de type raie.	88
3.33	Nage d'un poisson de type raie sur une p�riode de battements T	88
3.34	Sillage g�n�r� par la nage d'un poisson de type raie (iso-vorticit�).	89
3.35	Profil d'un poisson de type raie.	89
3.36	Nage d'un poisson de type raie sur une p�riode de battements T	89
3.37	Sillage g�n�r� par la nage d'un poisson de type knifefish (iso vorticit�).	90
3.38	Etapes de squelettisation : photos de profil et de dessus (gauche), segmentation et extraction du contour (milieu), construction de la level set (droite).	91
3.39	Redressement du contour (vue de dessus).	91
3.40	Exemples de reconstruction du profil 3D. A gauche : profil non d�form�. A droite : profil d�form�.	91
3.41	Visualisation de la densit� utilis�e. Le z�ro de la level set correspond au trait blanc ext�rieur. Coupe 2D, plan $z = 0$	93
3.42	Reconstruction d'une image pour $\tau = 0.5\Delta t_s$ (rouge) entre ρ_0 (vert) et ρ_1 (bleu). Coupe 2D, plan $z = 0$	93
3.43	Repr�sentation sch�matique des domaines utilis�s pour le calcul des forces et moments.	95
3.44	Comparaison des vitesses obtenues dans le cas I (suivi Lagrangien, champ dans le poisson) et dans le cas II (transport optimal, repr�sent�e uniquement � l'ext�rieur du corps).	95
3.45	Comparaison des vitesses obtenues par le poisson "lagrangien" et le poisson reconstruit � partir d'images.	96
3.46	Simulation de la nage d'un maquereau reconstruit � partir de photographies.	96
4.1	Exemple de simulation d'une h�lice de bateau. Photo d'une h�lice (gauche), mod�lisation de l'h�lice (centre) et simulation (droite, iso vorticit� � $Re = 10^4$).	100
4.2	Souff�rie et �olienne du NREL. http://wind.nrel.gov/amestest/	101
4.3	Mod�lisation de la pale du NREL. Twist from 30° at hub to -2.5° at tip.	101
4.4	Domaine de calcul 2D	102
4.5	Domaine de calcul avec un solveur lin�aire implicite	103
4.6	Twist initial ϕ et twist modifi� $\phi + \phi'$	104
4.7	Comparaison des r�sultats obtenu num�riquement avec les r�sultats exp�rimentaux.	105
4.8	Visualisation d'une pale d�velopp�e par la soci�t� enercon.	105
4.9	Visualisation de la pale initiale (NREL, � gauche), et de la pale optimis�e (� droite). Le mod�le simplifi� pr�dit 7% d'augmentation de puissance.	106
4.10	Mod�lisation de l'�olienne	106
4.11	Champ de vorticit� cr�e par l'�olienne du NREL obtenu num�riquement. Vitesse du vent en entr�e : 10m/s; vitesse de rotation 72 tours/minutes.	107
4.12	Mise en �vidence du tourbillon de bout de pale par un lâch� de fum�e.	108
4.13	Evolution de la tra�n�e et de la puissance en fonction de la vitesse du vent.	108
4.14	Evolution du coefficient de puissance c_p en fonction du <i>tip speed ratio</i> λ	109
4.15	Serpent de mer Pelamis. Cr�dit : Pelamis Wave Power.	109
4.16	Repr�sentation de la poutre dans le plan $z = 0$	111
4.17	Configuration et discr�tisation de la poutre �lastique.	111
4.18	Evolution temporelle de la vitesse du poisson avec loi de nage impos�e.	115
4.19	Nage passive d'un poisson d'apr�s Beal <i>et al.</i> (2006).	116
4.20	Poisson �lastique dans un sillage (champ de pression) pour $\mu_f = 10^{-6}$ et $E = 1MPa$ et $\mu_s = 0,01Pa/s$	116
4.21	Poisson �lastique dans un sillage (champ de pression) pour $\mu_f = 10^{-5}$ et $E = 0.1MPa$ et $\mu_s = 0,1Pa/s$	116
4.22	Poisson �lastique dans un sillage (champ de pression) pour $\mu_f = 10^{-5}$ et $E = 1MPa$ et $\mu_s = 0,1Pa/s$	117
4.23	Simulation num�rique de l'instabilit� de Rayleigh-Taylor (vorticit� � gauche et densit� � droite). $\rho_{haut} = 3, \rho_{bas} = 1$	118
4.24	Simulation de la chute d'une goutte d'eau.	118
4.25	Exemple de simulation d'un bateau. La longueur du bateau est 2m et sa vitesse est 2m/s. La ligne triple se d�place sur la coque.	119
4.26	Sillage g�n�r� par un bateau. La longueur du bateau est 1m et sa vitesse est 2m/s.	119

4.27 Déformation de la structure élastique "rigide" sous l'effet d'une vague déferlante.	120
4.28 Déformation de la structure élastique "molle" sous l'effet d'une vague déferlante.	120

Liste des tableaux

1.1	Description des différents modèles réduits. N_r est le nombre de modes POD utilisés pour construire le modèle réduit POD	20
1.2	Bases POD utilisées.	34
1.3	Erreur de projection L^2 ($\times 10^4$). L'erreur moyenne P_T est évaluée sur les 12 points P_i . Les bases $B_{Uniform}$ et B_{Subopt} sont calculées avec 160 snapshots. La base $B_{Initial}$ est calculée avec 80 snapshots.	34
2.1	Comparaison entre les vitesses terminales de la sédimentation de sphère pour plusieurs diamètres et viscosités du fluide.	66
3.1	Evolution de la vitesse moyenne \bar{u} , de la déviation relative d et du temps CPU t_{CPU} en fonction du pas d'espace h	73
3.2	Paramètres utilisés pour modéliser les quatre modes de nage.	75
3.3	Comparaison de quelques caractéristiques de nage pour $Re = 10^3$ et $Re = 10^4$	76
3.4	Comparaison de l'énergie $E^{(k)}$ dépensée pour nager sur une distance $d = 9$ pour $Re = 10^3$ et $Re = 10^4$. Les poissons F_1^r, F_2^r, F_3^r ont régulé l'amplitude de la queue pour nager à la vitesse du poisson F_4	77
3.5	Efficacité R de la nage intermittente pour F_1 à $Re = 10^3$ et $Re = 10^4$ avec différents paramètres $U_f = \alpha_f U_{max}$ et $U_i = \alpha_i U_{max}$	78
3.6	Pourcentage d'énergie économisée par le groupe de trois poissons par comparaison avec trois nageurs indépendants à $Re = 10^3$	79

Introduction générale

Position du problème

Mes domaines de recherche s'articulent autour de la modélisation et de la simulation numérique de systèmes complexes multi physiques avec en particulier des applications en mécanique des fluides.

D'un point de vue général, on peut nommer complexe tout système dans lequel interviennent de nombreux effets : mécaniques, thermiques, électromagnétiques, chimiques, biologiques etc., ...ces effets étant couplés de façon non linéaire à des échelles différentes. La modélisation, la prédiction, l'expérimentation pour de tels systèmes présentent des difficultés. La problématique des systèmes complexes est amenée à se développer en tant que telle non seulement dans le secteur des Sciences pour l'Ingénieur, mais aussi dans des disciplines comme la biologie et l'économie.

Les problèmes de l'expérimentation. Le couplage des effets rend l'expérimentation difficile et presque nécessairement indirecte. Il est, pour ces systèmes, délicat (à cause du couplage) d'isoler un paramètre et de le faire varier seul pendant une expérience. L'expérimentation est, quelquefois, impossible (en économie elle l'est toujours). Les paramètres des modèles ne peuvent être identifiés que par des méthodes indirectes faisant intervenir les modèles (méthodes dites inverses). La plupart des données accessibles ne peuvent être considérées que comme des grandeurs « contrôlables » et non pas mesurables. La métrologie est associée presque toujours à un problème inverse qui lui même se formule comme un problème de contrôle.

La modélisation. Un modèle de système complexe doit être maniable ou utilisable pour se faire la réduction et l'expertise de ces systèmes est nécessaire. La réduction consiste à trouver un système plus "simple" ayant, dans une gamme de paramètres donnés, le même comportement. L'expertise ayant pour objet de chercher les bonnes gammes et échelles des paramètres. Ces deux activités sont liées. La solution idéale au problème de la réduction est de trouver une solution analytique au modèle complet considéré. Ceci s'avère tout à fait impossible en général : on va donc plutôt chercher des "modèles approchés" dont on puisse maîtriser la solution et dont le comportement soit proche. Il y a, entre autres, deux approches adaptées à la réduction : les méthodes asymptotiques et d'invariance et les méthodes de projection. Dans la première il s'agit de simplifier le modèle et ses équations, dans la deuxième on simplifie l'espace de phase dans lequel varie les paramètres en n'autorisant les grandeurs dépendantes à n'évoluer que dans un espace de dimension finie.

La simulation. Le troisième volet est celui de la simulation. Etant donné les difficultés expérimentales et la lourdeur des modèles, l'étape de simulation est incontournable. La simulation doit être considérée comme une "expérience numérique" qui quelquefois remplacera l'expérience physique réelle.

Domaines de recherches

Principalement, je me suis intéressé à des problèmes issus de la mécanique des fluides. Après ma thèse j'ai également abordé des problèmes de modélisation, de contrôle et de simulation de systèmes de réaction diffusion avec une application à la lutte contre les feux de forêts. Cette thématique ne sera pas présentée dans ce mémoire.

Mes activités s'articulent principalement autour des points **modélisation** et **simulation** mentionnés précédemment. En particulier, je me suis intéressé à la réduction de modèle par méthode d'invariance (non présentée dans ce manuscrit) et essentiellement à la réduction de modèle sur bases réduites construites par Décomposition Orthogonale aux valeurs Propres (POD, *Proper Orthogonal Decomposition*). Mes activités ont

ensuite été centrées autour de la simulation numérique en mécanique des fluides pour des écoulements incompressibles. Cette thématique s'étend actuellement à des problèmes multi physiques tels que les interactions avec des structures élastiques.

Organisation du manuscrit

Ce manuscrit s'organise autour de deux axes de recherches : la réduction de modèle sur des bases réduites (partie I) et la modélisation et la simulation numérique en mécanique des fluides (partie II). Uniquement les activités de recherches engagées depuis mon arrivée à l'INRIA Bordeaux Sud-Ouest (ex-INRIA Futurs) seront présentées.

La première partie est composée d'un unique chapitre qui regroupe mes contributions à l'amélioration des modèles réduits construits par POD. Ces activités correspondent aux deux années après mon recrutement à INRIA, ainsi que des activités plus récentes dans le cadre du projet européen FP7 FFAST¹. En particulier, je me suis intéressé à améliorer la robustesse des fonctions de bases POD ainsi qu'à améliorer la stabilité des modèles réduits construits sur ces bases. Les fonctions de bases sont soit mises à jours via des algorithmes efficaces basés sur les résidus des équations détaillées lorsque des paramètres du système évoluent, soit des fonctions de bases représentatives d'une large gamme de paramètres sont générées par des méthodes inspirées des tessellations de Voronoi. Les modèles réduits sont stabilisés par des méthodes de type SUPG et VMS. Couplés à des simulations numériques directes (DNS), les modèles réduits peuvent permettre, dans certains cas, d'accélérer les calculs.

La seconde partie consacrée à la modélisation et la simulation numérique en mécanique des fluides s'articule autour de trois chapitres, les chapitres 2, 3 et 4. Cette activité a été engagée dans l'ANR CARPEiNTER² fin 2009. Un problème majeur pour la modélisation et simulation en mécanique des fluides est la gestion des interfaces (fluide/fluide, fluide/structure, ou encore fluide/fluide/structure). Nous avons choisi de résoudre tous les modèles par des méthodes eulériennes sur des maillages cartésiens. En particulier nous avons opté pour le modèle continu de Navier-Stokes. Le chapitre 2 présente les modèles eulériens utilisés et les méthodes numériques associées. Essentiellement ce chapitre s'articule autour de l'utilisation de grilles cartésiennes couplées avec des méthodes de pénalisation volumique/frontières immergées (pour prendre en compte les obstacles) et fonctions level set (pour suivre et capturer les interfaces). J'ai développé le code de calcul massivement parallèle NaSCar (Navier-Stokes Cartésien) dans des versions 2D et 3D qui regroupe toutes les méthodes précédemment citées. Le chapitre 3 est centré autour de l'étude de prototypes de nageurs autopropulsés de type poissons. Ceci est une application de l'interaction entre un fluide et un obstacle déformable qui se déplace dans un fluide. Le premier objectif de chapitre est donc d'illustrer le potentiel des méthodes développées au chapitre 2 pour modéliser et simuler des écoulements autour de géométries complexes. Le second objectif a été d'étudier la physique de la nage avec en particulier le souci de s'intéresser à la l'énergie dépensée. Nous avons commencé avec des simulations bidimensionnelles pour finalement arriver à des simulations tridimensionnelles plus réalistes avec notamment la simulation de la nage d'un poisson reconstruit à partir de photographies. Enfin, le chapitre 4 présente des études qui sont encore dans un stade exploratoire. Ce chapitre présente en partie quelques perspectives. Nous nous intéressons à des systèmes de l'extraction de l'énergie d'un écoulement. Nous présenterons essentiellement deux applications. La première application concerne l'étude des éoliennes. Cette application est apparue avec une collaboration avec la société VALEOL (localisée à Bègles en Gironde) filière du groupe VALOREM. En particulier, VALEOL est intéressé par l'optimisation des pales, et principalement d'étudier le potentiel bénéfique d'ajout d'appendices sur des pales existantes. Ce problème d'optimisation n'est pas réalisable avec l'outil NaSCar en raison des coûts de calcul (temps CPU et mémoire si une méthode adjointe était choisie). Nous avons alors développé un modèle réduit bidimensionnel axisymétrique basé sur la méthode de disque actif. Ce modèle est développé dans la thèse de CIFRE de Xin Jin. L'idée est alors d'optimiser la forme de la pale avec ce modèle simplifié, puis d'expertiser cette pale avec le modèle détaillé NaSCar. Pour le moment, le code de calcul détaillé n'est pas en mesure de simuler fidèlement ce type d'écoulement et doit être amélioré dans ce sens. La seconde application prospective est l'étude des *serpents de mer* qui sont des structures élastiques articulées qui flottent sur l'océan et qui devraient permettre de récupérer de l'énergie de la houle. Cette application est une interaction *two way* entre eau, air et une structure élastique. Ce dernier chapitre représente en soi une perspective des travaux à court et moyen terme. Le chapitre 5 présentent les conclusions et perspectives générales de ces travaux. Outre la perspective

1. Future Fast Aeroelastic Simulation Technologies, [texttthttp://www.bris.ac.uk/aerodynamics-research/ffast/](http://www.bris.ac.uk/aerodynamics-research/ffast/)

2. Cartesian grids, penalization and level set for the simulation and optimisation of complex flows, <http://www.math.u-bordeaux1.fr/CARPEiNTER/>

de continuer les travaux engagés sur la modélisation et la simulation de systèmes d'extraction d'énergie, une part importante des travaux futurs va se reconcentrer sur les modèles réduits. La thématique porte toujours sur l'énergie, mais cette fois-ci sur le stockage d'énergie. Nous allons encadrer une thèse avec Mejdi Azaiez (I2M, Bordeaux) et Angelo Iollo (IMB, Bordeaux), financée par le cluster CPU du labex, sur la modélisation et la simulation du stockage d'énergie en partenariat avec la société espagnole ABINGOA localisée à Seville. L'idée du stockage envisagé est par changement de phase du sel (dégagement de chaleur) contenu dans une matrice porteuse. De plus, nous avons donné une part importante aux modèles réduits dans la nouvelle équipe INRIA que nous sommes en train de créer (la branche "fluide" de l'équipe INRIA actuelle MC2).

Première partie

Réduction de modèle et bases réduites

Introduction et motivations de l'étude

Depuis le début des années 1980 l'optimisation aérodynamique et aéroacoustique des véhicules terrestres et des avions a été approchée numériquement. Les applications concernent principalement les écoulements turbulents à très hauts nombres de Reynolds. Les simulations numériques et le contrôle de ce type d'écoulements nécessitent des ressources informatiques de très grandes capacités (clusters massivement parallèles couplés avec beaucoup de mémoire). En effet, après discrétisation des équations de Navier-Stokes nous devons résoudre un système d'équations dont la complexité croît de façon algébrique avec le nombre d'inconnues à résoudre. De nos jours, et malgré les progrès conséquents apparus ces dernières années dans le domaine du calcul (puissance des ordinateurs et nouveaux algorithmes de résolution), il est encore très difficile de résoudre des problèmes de grandes dimensions en temps réel, ce qui est *in fine*, un enjeu majeur pour les industriels.

Pour éviter ces difficultés, il est possible de déterminer des modèles réduits de la dynamique en ne conservant qu'un nombre limité de modes globaux. Le choix de ces modes n'est pas unique et dépend fortement des caractéristiques de l'écoulement que l'on souhaite approcher. Ces modes peuvent également dépendre de certaines quantités de sortie que l'on souhaite représenter au mieux (Goal-oriented models, Bui-Thanh *et al.* (2007)), comme la traînée générée par un obstacle par exemple. Plusieurs méthodes sont habituellement utilisées, parmi lesquelles la décomposition orthogonale aux valeurs propres (*Proper Orthogonal Decomposition, POD*) (Holmes *et al.*, 1996; Sirovich, 1987; Cordier et Bergmann, 2002a), la *balanced truncation* (Nocedal et Wright, 1999; Rowley, 2005; Willcox et Peraire, 2002), les modes globaux de stabilité (Akervik *et al.*, 2007), les modes de Galerkin (Noack et Eckelman, 1994) *etc.* En raison de l'optimalité énergétique de la base, nous avons opté pour la POD. Avec cette technique il est possible d'extraire les caractéristiques dominantes (modes POD) d'une base de données, et finalement d'obtenir un modèle réduit (*Reduced Order Model, ROM*) par projection de Galerkin des équations du modèle sur la base POD. Bien que cette technique puisse être très efficace dans certaines situations d'écoulement, elle présente également quelques défauts. En plus de l'éventuel manque de stabilité numérique des méthodes POD/Galerkin, même pour des systèmes simples (Rempfer, 2000), les principaux défauts sont :

- Puisque dans la plupart des applications pour les écoulements incompressibles les base POD sont construites à partir des champs de vitesse³ il est nécessaire de modéliser le terme de pression. Généralement, dans la plupart des écoulements fermés la contribution du terme de pression s'annule en choisissant des conditions aux limites appropriées. Cependant, pour des écoulements qui présentent des couches de cisaillement instables telles que des couches de mélange, il a été prouvé par Noack *et al.* (2005) que négliger le terme de pression pouvait engendrer des erreurs de grandes amplitudes dans le modèle de Galerkin. Une modélisation précise de ce terme de pression est alors indispensable pour représenter fidèlement ce type d'écoulements (Noack *et al.*, 2005; Galletti *et al.*, 2004). Afin de passer outre cette difficulté de modélisation, nous avons développé un modèle réduit étendu à la pression §1.1.3. Le terme de pression peut alors directement être calculé à partir du mode POD de pression.
- En raison de l'optimalité énergétique de la base POD, un nombre très limité de fonctions de base POD est suffisant pour représenter la majeure partie de l'énergie cinétique de l'écoulement⁴. Dans l'optique de la réduction de modèle, nous conservons uniquement les modes associés aux structures d'écoulement à grandes échelles (telles que les tourbillons de l'allée de Von Kármán qui se développe en particulier dans le sillage de corps). Mais puisque la majeure partie de la dissipation visqueuse s'effectue dans les petites structures représentées par les modes qui ne sont pas considérés dans le modèle réduit (car pas suffisamment énergétiques), le modèle réduit n'est pas en mesure de dissiper suffisamment d'énergie. Il est alors nécessaire de développer un modèle de fermeture afin de modéliser les interactions entre les structures calculées par le POD ROM et les structures non résolues. Ce problème est similaire à celui de la Simulation des Grandes Echelles (*Large Eddy Simulation, LES*, Sagaut (2005)) d'écoulements turbulents. Dans cette étude le modèle réduit est fermé en utilisant les résidus des équations de Navier-Stokes en exploitant des idées proches des méthodes SUPG (*Streamline Upwind Petrov-Galerkin*) et VMS (*Variational Multiscale*, Bazilevs *et al.* (2007)).
- Enfin, puisque les fonctions de bases POD sont optimales pour représenter les caractéristiques dominantes incluses dans la base de données générée pour certains paramètres d'entrée, la même base n'a aucune garantie de représenter fidèlement un écoulement généré avec d'autres paramètres d'entrée (Prabhu *et al.*, 2001; Noack *et al.*, 2003; Bergmann et Cordier, 2008). Pour répondre à cette

3. Dans la plupart des travaux expérimentaux le champ de pression n'est pas directement accessible.

4. Ceci est vrai pour de simples écoulements bidimensionnels, mais des milliers de modes peuvent être nécessaires pour représenter fidèlement l'énergie fluctuante des écoulements turbulents pleinement développés.

problématique, nous proposons plusieurs méthodes pour adapter la base POD pour des coûts de calculs réduits. Ceci est le point central de la section §1.3.

Ce chapitre est essentiellement basé sur la publication Bergmann *et al.* (2009), et est complété par les travaux réalisés dans le projet européen FP7 FFAST piloté par l'université de Bristol (UK) avec Airbus UK. Ces travaux dans le projet européen ont été réalisés en collaboration avec Edoardo Lombardi et Haysam Telib d'OPTIMAD Engineering⁵, spin off du Politecnico de Torino (Bergmann *et al.*, 2012a,b, 2013).

5. OPTimization Methods for Aerodynamics Design, <http://www.optimad.it>

Chapitre 1

Contributions aux modèles réduits basés sur la Décomposition Orthogonale aux valeurs Propres

Sommaire

1.1	Du modèle classique vers le modèle à pression	9
1.1.1	Configuration d'étude	9
1.1.2	Modèle réduit standard basé sur la décomposition orthogonale aux valeurs propres	10
1.1.3	Un modèle POD avec pression	12
1.2	Amélioration de la stabilité du modèle réduit	17
1.2.1	Synthèse des méthodes de stabilisation existantes	17
1.2.2	Méthodes de stabilisation basées sur le résidu : Modèle $B^{[N_r;K]}$	19
1.2.3	Méthodes SUPG et VMS : modèles $C^{[N_r]}$ et $D^{[N_r]}$	19
1.2.4	Résultats numériques des méthodes de stabilisation	20
1.3	Amélioration de la robustesse de la base POD	21
1.3.1	Une méthode basée sur les sous espaces de Krylov	23
1.3.2	Une méthode hybride DNS/POD ROM	26
1.3.3	Construction d'une base POD robuste : Iterative Centroidal Voronoi Tessellation (ICVT)	30
1.4	Conclusions et perspectives	34

1.1 Du modèle classique vers le modèle à pression

1.1.1 Configuration d'étude

Dans une grande partie de cette étude, nous nous intéressons à l'écoulement bidimensionnel confiné autour d'un cylindre carré (figure 3.1). Cet écoulement est intéressant car il présente un détachement de la couche limite, un sillage ainsi que des interactions avec les parois. Nous étudions les équations de Navier-Stokes dans leur forme sans dimension :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \Delta \mathbf{u}, \quad (1.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (1.1b)$$

où le nombre de Reynolds $Re = U_\infty L / \nu$ est construit avec la vitesse d'entrée maximale $U_\infty = u(0, H/2)$, la longueur d'un côté du carré L et la viscosité cinématique du fluide ν . Dans ce qui suit nous considérons $Re = 100$ et $Re = 200$, *i.e.* le régime laminaire. Les mêmes paramètres et méthodes numériques que ceux

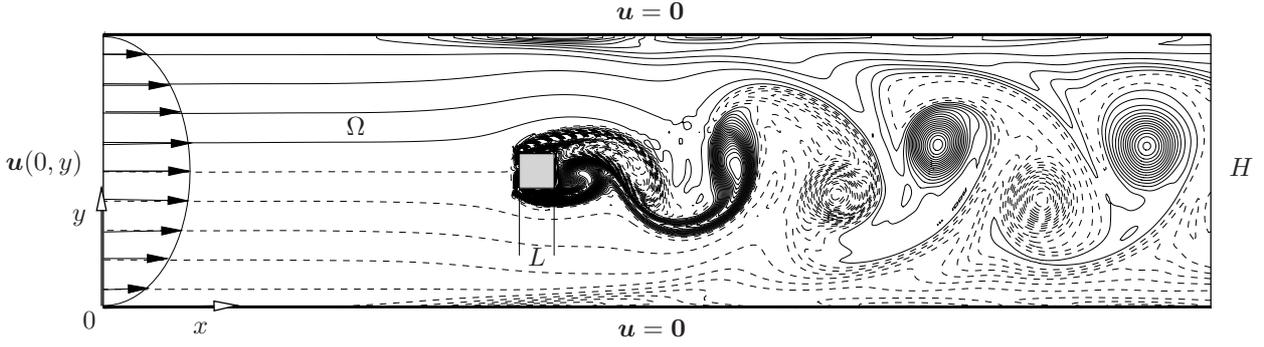


FIGURE 1.1 – Configuration d'écoulement et cliché de la vorticité pour $Re = 200$.

présentés par Galletti *et al.* (2004) sont utilisés. En particulier, le blocage $\beta = L/H$ est égal à 0.125 et le domaine Ω est $(0, 4H) \times (0, H)$. Un cliché de la vorticité est présenté sur la figure 3.1. Les phénomènes décrits précédemment sont visibles. Ces résultats numériques ont été obtenus grâce au code de calcul NS2D¹ développé à l'institut de Mathématiques de Bordeaux par Charles-Henri Bruneau.

1.1.2 Modèle réduit standard basé sur la décomposition orthogonale aux valeurs propres

Décomposition orthogonale aux valeurs propres

La décomposition orthogonale aux valeurs propres (*Proper Orthogonal Decomposition, POD*) a été introduite initialement par Lumley (1967) en 1967 pour définir les structures cohérentes présentes dans un écoulement turbulent. Des études détaillées de la POD peuvent être trouvées dans Holmes *et al.* (1996), Sirovich (1987) et Cordier et Bergmann (2002a). La POD, également connue sous le nom de décomposition de Karhunen-Loève, analyse en composantes principales, méthode de décomposition empirique, consiste à rechercher la fonction déterministe $\Phi(\mathbf{x})$ qui est le mieux corrélée en moyenne aux réalisations de l'écoulement $\mathbf{U}(\mathbf{x}, t)$. Les réalisations $\mathbf{U}(\mathbf{x}, t)$ sont généralement les champs de vitesse, mais nous verrons comment les étendre au champ de pression. Puisque les données sont issues de simulations numériques, nous adoptons la méthode des snapshots introduite par Sirovich (1987) (voir Cordier et Bergmann (2002a) pour plus de justifications). Le problème d'optimisation avec contraintes se réduit à un problème aux valeurs propres pour une équation intégrale de Fredholm :

$$\int_0^T C(t, t') a_n(t') dt' = \lambda_n a_n(t) \quad (1.2)$$

où le tenseur de corrélation $C(t, t')$ est :

$$C(t, t') = \frac{1}{T} (\mathbf{U}(\mathbf{x}, t), \mathbf{U}(\mathbf{x}, t'))_{\Omega}. \quad (1.3)$$

Le produit scalaire $(\cdot, \cdot)_{\Omega}$ entre deux champs \mathbf{U} et \mathbf{V} est :

$$(\mathbf{U}, \mathbf{V})_{\Omega} = \int_{\Omega} \mathbf{U} \cdot \mathbf{V} d\mathbf{x} = \int_{\Omega} \sum_{i=1}^{n_c} U^i V^i d\mathbf{x},$$

où U^i est la i^{me} composante du vecteur \mathbf{U} de dimension n_c .

Les valeurs propres λ_n ($n = 1, 2, \dots$) déterminées par (1.2) sont toutes positives et forment une suite décroissante et convergente. Chaque valeur propre correspond à la contribution du mode Φ_n à l'information relative des données initiales. Si \mathbf{U} est le champ de vitesse, l'information relative est l'énergie cinétique.

1. Cette étude est antérieure au développement du code de calcul NaSCar.

Dans l'équation (1.2), les termes a_n sont les modes propres temporels POD d'ordre n . Ces modes forment une base orthonormée et vérifient :

$$\frac{1}{T} \int_0^T a_n(t) a_m(t) dt = \lambda_n \delta_{nm}. \quad (1.4)$$

Les vecteurs propres associés Φ_n forment une base complète et vérifient : $(\Phi_n, \Phi_m)_\Omega = \delta_{nm}$.

Les fonctions de bases Φ_n^i peuvent être calculées à partir des réalisations U^i et des coefficients a_n avec :

$$\Phi_n^i(\mathbf{x}) = \frac{1}{T \lambda_n} \int_0^T U^i(\mathbf{x}, t) a_n(t) dt. \quad (1.5)$$

Puisque les fonctions de base POD sont une combinaison linéaire des réalisations de l'écoulement, elles ont les mêmes propriétés. Par exemple, si la base est construite à partir de champs de vitesse à divergence nulle, les fonctions de base seront également à divergence nulle. De plus, les fonctions de bases satisfont les mêmes conditions aux limites.

Le jeu de modes POD $\{\Phi_n\}_{n=1}^{N_{POD}}$ est complet dans le sens où toute réalisation $\mathbf{U}(\mathbf{x}, t)$ contenue dans les données initiales peut être décomposée avec une précision arbitraire (en fonction de $N_{POD} \geq 1$) sur les fonction de bases

$$\mathbf{U}(\mathbf{x}, t) \simeq \widehat{\mathbf{U}}^{[1, \dots, N_{POD}]}(\mathbf{x}, t) = \sum_{n=1}^{N_{POD}} a_n(t) \Phi_n(\mathbf{x}). \quad (1.6)$$

Nous notons $\widehat{\mathbf{U}}^{[1, \dots, N_{POD}]}$ une estimation de \mathbf{U} , où les crochets contiennent les indices des modes utilisés. Dans la suite nous considérons N_s réalisations de l'écoulement (appelées snapshots) $\mathbf{U}(\mathbf{x}, t_i)$, $\mathbf{x} \in \Omega$, extraites à $t_i \in [0, T]$, $i = 1, \dots, N_t$.

L'optimalité énergétique de la base POD suggère qu'un faible nombre de modes est suffisant pour décrire de façon efficace les snapshots de la base de données *i.e.* $N_r \ll N_s$. En pratique, N_r est généralement déterminé comme étant le plus petit entier M tel que l'information relative (*Relative Information Content*) $RIC(M) = \sum_{i=1}^M \lambda_i / \sum_{i=1}^{N_s} \lambda_i$, est plus grande qu'un certain pourcentage prédéfini, δ . Donc $N_{POD} = N_r$ et l'approximation (1.6) devient :

$$\mathbf{U}(\mathbf{x}, t) \simeq \widehat{\mathbf{U}}^{[1, \dots, N_r]}(\mathbf{x}, t) = \sum_{n=1}^{N_r} a_n(t) \Phi_n(\mathbf{x}). \quad (1.7)$$

Modèle réduit classique : avantages et inconvénients

Un modèle réduit classique est construit à partir des champs de vitesse de l'écoulement, $\mathbf{U}(\mathbf{x}, t) \equiv \mathbf{u}(\mathbf{x}, t)$. La décomposition (1.7) devient alors :

$$\mathbf{u}(\mathbf{x}, t) \simeq \sum_{n=1}^{N_r} a_n(t) \phi_n(\mathbf{x}), \quad (1.8)$$

où ϕ_n est le n -ième vecteur de base POD. Un système dynamique d'ordre réduit est obtenu à l'aide d'un projection de Galerkin des équations de Navier-Stokes (1.1). Cela consiste à restreindre la forme faible des équations de Navier-Stokes au sous espace $\mathcal{S}_{N_r}^{POD}$ engendré par les premières N_r fonctions de bases spatiales ϕ_n .

Finalement, la projection de Galerkin s'écrit :

$$\left(\phi_i, \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right)_\Omega = - (\phi_i, \nabla p)_\Omega + \left(\phi_i, \frac{1}{Re} \Delta \mathbf{u} \right)_\Omega. \quad (1.9)$$

Le terme de pression $(\phi_i, \nabla p)_\Omega$ est généralement négligé car il ne peut pas être évalué avec la formulation standard en vitesse. En utilisant la décomposition (1.8), Le modèle réduit devient (voir Cordier et Bergmann (2002b) pour plus de détails) :

$$\frac{da_i(t)}{dt} = \mathcal{A}_i + \sum_{j=1}^{N_r} \mathcal{B}_{ij} a_j(t) + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} \mathcal{C}_{ijk} a_j(t) a_k(t) \quad i = 1, \dots, N_r. \quad (1.10a)$$

avec les conditions initiales

$$a_i(0) = (\mathbf{u}(\mathbf{x}, 0), \boldsymbol{\phi}_i(\mathbf{x}))_{\Omega} \quad i = 1, \dots, N_r. \quad (1.10b)$$

Un problème bien connu de ce type de modèle réduit est que l'intégration numérique de ce modèle (1.10) donne une solution qui a tendance à s'éloigner de la solution du modèle détaillé (Navier-Stokes). Ce phénomène ne permet donc pas de calculer une solution précise aux temps longs (Sirisup et Karniadakis, 2004). Dans certains cas, même la solution aux temps courts n'est pas en mesure de fournir une solution approchée suffisamment précise.

Nous pouvons identifier trois sources d'erreurs numériques. Tout d'abord, comme il vient d'être mentionné, un modèle réduit POD/Galerkin peut être instable structurellement même pour des problèmes simples (Rempfer, 2000). Ensuite, le terme de pression est souvent négligé dans un modèle réduit POD (POD ROM). Nous verrons au paragraphe §1.1.3 qu'il est possible de créer un modèle réduit étendu avec le champ de pression. Enfin, la dernière source d'erreur est due à la troncature effectuée dans l'approche POD/Galerkin. En effet, seules les structures les plus énergétiques sont calculées. La dissipation visqueuse s'effectue principalement dans les petites et moyennes structures qui ne sont pas calculées. Le système n'est pas en mesure de dissiper assez d'énergie et peut devenir instable ou converger vers une solution erronée. Ce problème est comparable à la simulation de grandes échelles (SGE, ou LES en anglais pour Large Eddy Simulation) ou le transfert entre les échelles résolues et les échelles de sous grille doit être modélisé (Sagaut, 2005). Par exemple, quatre modes sont suffisants pour retrouver plus de 99% de l'énergie cinétique du sillage d'un cylindre circulaire en régime 2D laminaire, mais la solution de ce modèle réduit ne converge pas vers la solution numérique du modèle détaillé de Navier-Stokes (Deane *et al.*, 1991). Il est alors nécessaire de stabiliser le POD ROM. Dans cette étude, nous étendons la base POD au champ de pression. Avec les champs de vitesse et de pression nous pouvons évaluer les résidus des équations détaillées (Navier-Stokes). Différentes stratégies de stabilisation seront basées sur ces résidus.

1.1.3 Un modèle POD avec pression

Il a été montré que la contribution du terme de pression s'annule pour des configurations d'écoulements fermés. Cependant, pour la plupart des écoulements avec au moins une condition aux limites ouverte (couche de mélange par exemple), ne pas prendre en compte ce terme de pression peut amener des erreurs importantes dans le modèle réduit (Noack *et al.*, 2005). Une solution peut être de modéliser ce terme Noack *et al.* (2005); Galletti *et al.* (2004), mais nous souhaitons le calculer de façon précise. Le but de ce paragraphe est de dériver un modèle réduit étendu à la pression de telle sorte que nous puissions reconstruire les champs de vitesse et de pression avec la solution du POD ROM. Ce terme de pression peut être calculé en utilisant $p = \tilde{p}$ (voir (1.11b)). Un autre point important est que la connaissance de la pression nous permettra de calculer de façon précise les résidus des équations de Navier-Stokes évalués avec la solution POD ROM². Les résidus de Navier-Stokes vont nous aider à stabiliser le modèle réduit (§1.2) et à améliorer la robustesse du sous espace POD (§1.3).

Construction du modèle réduit étendu à la pression

Comme il a déjà été mentionné §1.1.2, un modèle réduit est basé sur la restriction de la forme faible des équations de Navier-Stokes à vivre sur le sous espace $\mathcal{S}_{N_r}^{POD}$ engendré par les N_r premières fonctions de base $\boldsymbol{\Phi}_i$. Nous développons ici une base globale pour la vitesse et la pression (voir Bergmann (2004) pour plus de précisions et justifications). Les solutions \mathbf{u} et p sont approchés par :

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_{i=1}^{N_r} a_i(t) \boldsymbol{\phi}_i(\mathbf{x}) \quad (1.11a)$$

$$\tilde{p}(\mathbf{x}, t) = \sum_{i=1}^{N_r} a_i(t) \psi_i(\mathbf{x}). \quad (1.11b)$$

Les fonctions de base pour la vitesse et la pression $\boldsymbol{\phi}_i$ et ψ_i sont déterminées en utilisant $\mathbf{U}(\mathbf{x}, t) = (\tilde{\mathbf{u}}(\mathbf{x}, t), \tilde{p}(\mathbf{x}, t))^T$ pour calculer le tenseur des corrélations (1.3). Les fonctions de base $\boldsymbol{\phi}_i$ et ψ_i sont déterminées par $\boldsymbol{\Phi}(\mathbf{x}, t) = (\boldsymbol{\phi}(\mathbf{x}, t), \psi(\mathbf{x}, t))^T$, $\boldsymbol{\Phi}(\mathbf{x}, t)$ étant obtenue par (1.5). La substitution (1.11) dans les

2. Il est également possible d'évaluer les résidus de Navier-Stokes en formulation vortex.

équations de conservation de la quantité de mouvement (1.1a) donne :

$$\sum_{j=1}^{N_r} \phi_j \frac{da_j}{dt} + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} (\phi_j \cdot \nabla) \phi_k a_j a_k = - \sum_{j=1}^{N_r} \nabla \psi_j a_j + \frac{1}{Re} \sum_{j=1}^{N_r} \Delta \phi_j a_j. \quad (1.12)$$

Une projection de Galerkin de (1.12) donne :

$$\left(\phi_i, \sum_{j=1}^{N_r} \phi_j \frac{da_j}{dt} + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} (\phi_j \cdot \nabla) \phi_k a_j a_k + \sum_{j=1}^{N_r} \nabla \psi_j a_j - \frac{1}{Re} \sum_{j=1}^{N_r} \Delta \phi_j a_j \right)_{\Omega} = 0. \quad (1.13)$$

Le modèle réduit est alors :

$$\sum_{j=1}^{N_r} L_{ij}^{(m)} \frac{da_j}{dt} = \sum_{j=1}^{N_r} B_{ij}^{(m)} a_j + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} C_{ijk}^{(m)} a_j a_k, \quad (1.14)$$

où les coefficients³ L_{ij}^m , B_{ij}^m et C_{ijk}^m sont :

$$L_{ij}^{(m)} = + (\phi_i, \phi_j)_{\Omega}, \quad (1.15a)$$

$$B_{ij}^{(m)} = - \left(\phi_i, \frac{1}{Re} \Delta \phi_j - \nabla \psi_j \right)_{\Omega}, \quad (1.15b)$$

$$C_{ijk}^{(m)} = - (\phi_i, (\phi_j \cdot \nabla) \phi_k)_{\Omega}. \quad (1.15c)$$

L'exposant m est pour les équations de quantité de mouvement.

Dans ce modèle réduit nous avons utilisé en base globale construite par POD, mais cette méthode peut facilement être transposée à d'autres bases globales comme les modes de stabilité (Akervik *et al.*, 2007). De plus, il peut être intéressant d'utiliser des modes qui ne sont pas à divergence nulle comme les résidus des équations de Navier-Stokes. De tels modes peuvent être utilisés pour stabiliser le POD ROM §1.2 et pour améliorer la représentativité de sous espace POD 1.3. Si le modèle (1.14) n'est pas construit avec de modes à divergence nulle, il ne satisfait pas l'équation de continuité. Il est alors nécessaire d'ajouter une contrainte au modèle réduit.

Un modèle réduit modifié qui satisfait les équations de conservation de la quantité de mouvement et de la masse est obtenu en partant de la formulation faible des équations de Navier-Stokes :

$$\left(\mathbf{w}_i, \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla p - \frac{1}{Re} \Delta \mathbf{u} \right)_{\Omega} + (q_i, \nabla \cdot \mathbf{u})_{\Omega} = 0, \quad (1.16)$$

où \mathbf{w}_i et q_i appartiennent à des sous espaces appropriés. Les champs de vitesse et de pression sont alors représentés sur la base POD $\{\phi_i\}_{i=1}^{N_r}$ et $\{\psi_i\}_{i=1}^{N_r}$ en utilisant les équations (1.11a) et (1.11b).

Une solution est alors d'utiliser une projection de Galerkin où $\mathbf{w}_i = \phi_i$ et $q_i = \psi_i$. Une autre approche est d'utiliser $\mathbf{w}_i = \phi_i$ et $q_i = \alpha (\nabla \cdot \phi_i)^T$. Ce choix correspondrait à la minimisation du résidu de l'équation de continuité, $\sum_{j=1}^{N_r} a_j \nabla \cdot \phi_j$, au sens des moindres carrés. Pour $\alpha \gg 1$ nous avons :

$$\sum_{j=1}^{N_r} B_{ij}^c a_j = 0,$$

où $B_{ij}^{(c)} = (\nabla \cdot \phi_i)^T \nabla \cdot \phi_j$ et l'exposant c représente l'équation de continuité. Numériquement, cette dernière approche donne de meilleurs résultats, et le modèle réduit que nous utilisons est donc :

$$\sum_{j=1}^{N_r} L_{ij}^{(m)} \frac{da_j}{dt} = \sum_{j=1}^{N_r} \left(B_{ij}^{(m)} + \alpha B_{ij}^{(c)} \right) a_j + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} C_{ijk}^{(m)} a_j a_k \quad (1.17)$$

où le poids α doit être fixé par l'utilisateur. Dans cette étude nous avons choisi $\alpha = 10^{-2}$.

3. D'une façon générale, nous avons $(\Phi_i, \Phi_j)_{\Omega} = \delta_{ij}$, mais pas $(\phi_i, \phi_j)_{\Omega} = \delta_{ij}$. Donc, $L_{ij}^{(m)} \neq \delta_{ij}$.

Puisque nous utilisons la décomposition (1.11), le champ moyen est calculé et reconstruit à l'aide du modèle réduit $\bar{\mathbf{U}}(\mathbf{x}, t) = a_1(t)\Phi_1(\mathbf{x})$. Lors de la résolution numérique du modèle réduit, des variations du coefficient a_1 peuvent apparaître. Le débit est alors modifié. Il est donc nécessaire d'ajouter une deuxième contrainte au modèle réduit pour forcer le débit à rester constant (1.17). Pour un écoulement confiné 2D, la conservation du débit est :

$$\int_{\mathcal{S}} u \, ds = c, \quad (1.18)$$

où \mathcal{S} est la section verticale du canal et c est une constante donnée par la condition initiale. Numériquement, la conservation du débit doit être constante sur chaque section $\mathcal{S}_l \equiv \mathcal{S}(x_l)$, $1 \leq l \leq N_X$ où N_X est le nombre de points de discrétisation dans la direction x . Ceci est simplifié par l'utilisation d'un maillage cartésien.

Avec $\phi = (\phi^u, \phi^v)^T$ et (1.11a), la condition (1.18) est approchée par⁴ :

$$\sum_{i=1}^{N_r} a_i(t) \int_{\mathcal{S}_l} \phi_j^u \, ds = c, \quad (1.19)$$

La constante est initialement évaluée par projection d'un snapshot sur la base ϕ . Numériquement nous utilisons $c \equiv 1$. La conservation du débit est :

$$\sum_{j=1}^{N_r} \frac{da_j}{dt} \int_{\mathcal{S}_l} \phi_j^u \, ds = 0.$$

En notant \mathbf{f}_j le vecteur de composantes $f_j^l = \int_{\mathcal{S}_l} \phi_j^u \, ds$, la conservation du débit sur tout le domaine Ω s'écrit :

$$\sum_{j=1}^{N_r} \frac{da_j}{dt} \mathbf{f}_j = \mathbf{0}.$$

Cette contrainte supplémentaire est prise en compte en enrichissant le sous espace de projection avec $\beta \mathbf{f}_i$. Pour $\beta \gg 1$, nous avons au sens des moindres carrés :

$$\sum_{j=1}^{N_r} L_{ij}^r \frac{da_j}{dt} = 0,$$

où $L_{ij}^r = \mathbf{f}_i^T \mathbf{f}_j$. L'exposant r est pour la conservation du débit (*flow Rate conservation*). Finalement, le modèle réduit que nous allons utiliser est :

$$\sum_{j=1}^{N_r} \left(L_{ij}^{(m)} + \beta L_{ij}^{(r)} \right) \frac{da_j}{dt} = \sum_{j=1}^{N_r} \left(B_{ij}^{(m)} + \alpha B_{ij}^{(c)} \right) a_j + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} C_{ijk}^{(m)} a_j a_k \quad (1.20a)$$

avec les conditions initiales

$$a_i(0) = (\mathbf{U}(\mathbf{x}, 0), \Phi_i(\mathbf{x}))_{\Omega} \quad i = 1, \dots, N_r, \quad (1.20b)$$

où nous choisissons $\beta = 10^2$. Les paramètres α et β ont été optimisés sur notre cas test. Ce modèle réduit satisfait sous une forme faible les équations de conservation de la quantité de mouvement et de conservation de la masse, ainsi que la conservation du débit, et ce, même en cas d'utilisation de modes qui ne sont pas à divergence nulle.

Résultats numériques du modèle réduit complet

Le modèle réduit (1.20) est utilisé sur le cas test présenté à la section 1.1.1 pour $Re = 200$. La base POD Φ est construite par la méthode des snapshots introduite par Sirovich (1987). Nous utilisons ici 80 snapshots, répartis uniformément sur une période de lâché tourbillonnaire, pour calculer le tenseur de corrélations (1.3). Le spectre de valeurs propres est représenté sur la figure 1.2. Ce spectre est dégénéré dans le sens où il présente des paires de valeurs propres identiques pour les modes de fluctuation (le mode moyen est indexé par 1). La base POD est obtenue à l'aide d'une projection des valeurs propres du tenseur des corrélations sur le jeu de snapshots.

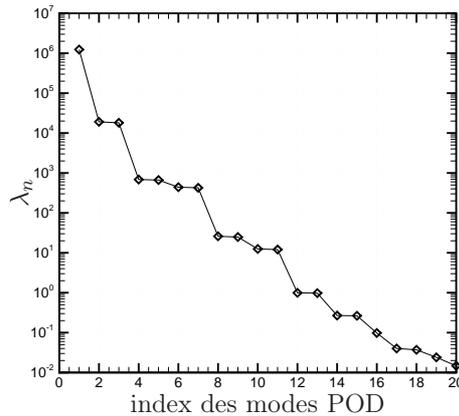


FIGURE 1.2 – Spectre de valeurs propres.

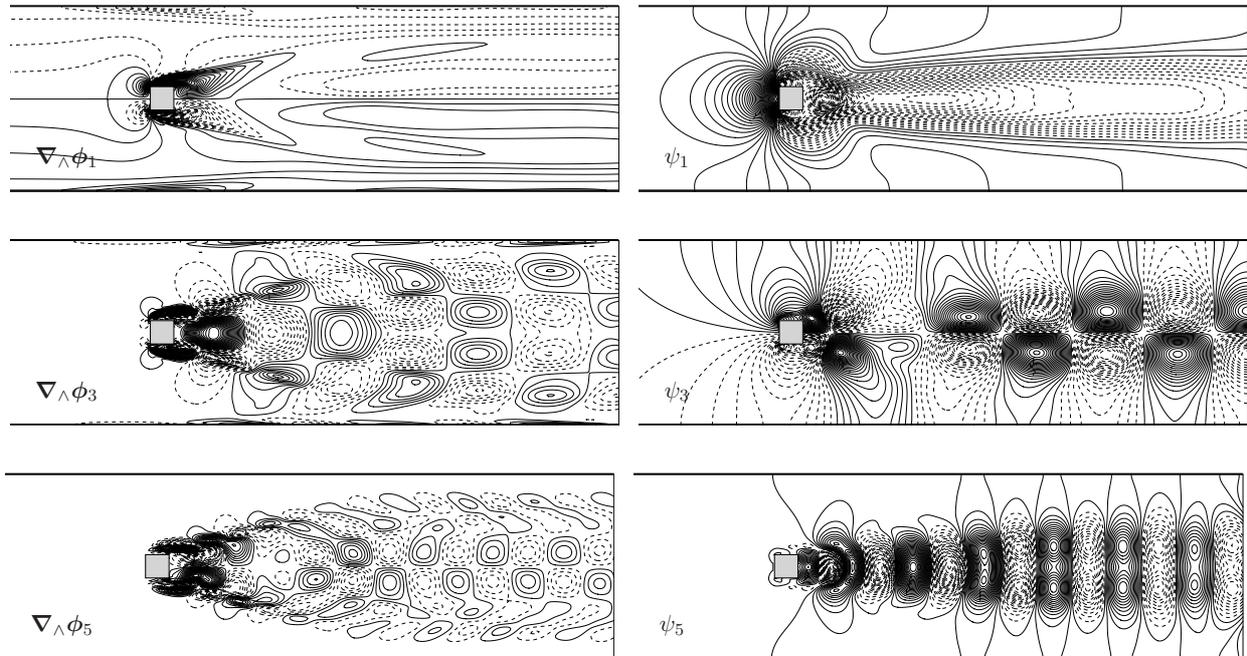


FIGURE 1.3 – Représentation de quelques modes POD en iso vorticité (gauche) et isobares (droite). Les traits pointillés représentent des valeurs négatives (la pression est arbitrairement initialisée à zéro).

Certains modes POD sont représentés sur la figure 1.3. L'évolution de l'information relative RIC introduite au §1.1.2 est présentée sur la figure 1.4. Nous voyons que seulement les 5 premiers modes sont suffisants pour représenter plus de 98% de l'information. Un choix raisonnable de ces 5 modes est de prendre les 5 premiers. Cependant, on peut voir que dans notre cas, on pourrait obtenir pratiquement la même information relative en remplaçant les modes 4 et 5 par les modes 6 et 7. En effet, même si structurellement ces

4. Si des modes POD sont utilisés, nous pouvons prendre $N_r = 1$ puisque la conservation du débit est uniquement donnée par l'écoulement moyen. Cependant, pour d'autres modes qui ne satisfont pas la conservation de débit, nous avons $N_r \neq 1$.

modes sont très différents (voir figure 1.3), ils contribuent de la même manière au contenu informatif (voir figure 1.5). Un choix judicieux n'est pas évident. Une autre méthode consiste à sélectionner tous les modes qui ont une contribution supérieure à un seuil à définir (voir figure 1.5). Ici, tous les modes ayant une contribution supérieure à 10^{-2} sont conservés, ce qui correspond à 10 modes de fluctuation plus le mode du champ moyen, *i.e.* $N_r = 11$ modes. Une fois que nous avons calculé les opérateurs du modèle réduit (1.20) à $N_r = 11$ modes,

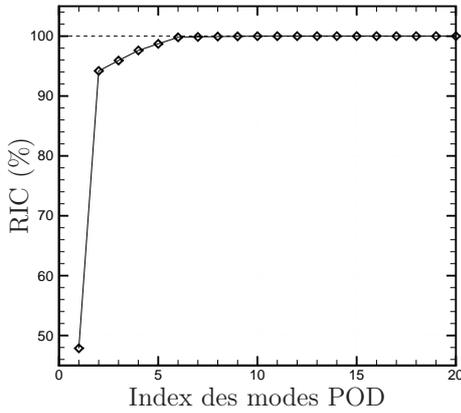


FIGURE 1.4 – RIC des modes de fluctuation.

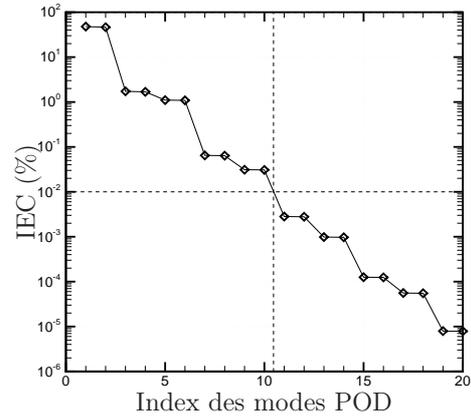


FIGURE 1.5 – Spectre des valeurs propres.

nous effectuons une prédiction sur plus de 1000 périodes de lâché tourbillonnaire. La figure 1.6 présente l'évolution temporelle de quelques coefficients $\{a_i\}_{i=1}^{N_r}$, solution du système (1.20), sur 40 périodes. Aucune divergence n'est observée. La figure 1.7 présentent une comparaison de quelques cycles limites obtenus par projection (DNS) et par prédiction (POD ROM). Les cycles limites prédits sont en très bon accord avec les cycles limites projetés, et ce, même pour les petites échelles à hautes fréquences où la dynamique est plus complexe. Le système (1.20) à 11 modes est numériquement stable pour une intégration aux temps longs et aucune calibration n'est nécessaire. Par contre, comme il a déjà été mentionné (§1.1.2), si l'on ne conserve

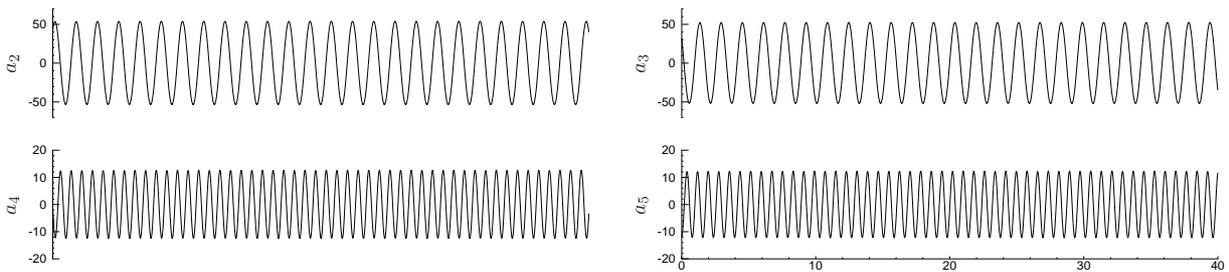


FIGURE 1.6 – Evolution temporelle des coefficient de prédiction sur 40 périodes de lâché tourbillonnaire pour le modèle à 11 modes.

pas suffisamment de modes nous pouvons observer une déviation de la solution qui peut même amener le système à diverger. En effet, nous voyons sur la figure 1.8 que la solution du système à 5 modes donne des cycles limites éronnés. De plus, la solution du système à 3 modes diverge (figure 1.9). Dans ce cas très simple, nous avons vu que $N_r = 11$ modes sont suffisants pour construire un modèle réduit qui soit stable. Cependant, dans la plupart des applications (3D, turbulence, géométries complexes, *etc*) le nombre de modes nécessaires peut devenir très grand. En général, approximativement 60% à 80% de l'énergie cinétique est retenue et de ce fait le ROM est instable. Dans ce qui suit, nous présentons des méthodes de stabilisation basées sur le résidu des équations de Navier-Stokes en utilisant un faible nombre de modes (3 ou 5 modes dans notre cas).

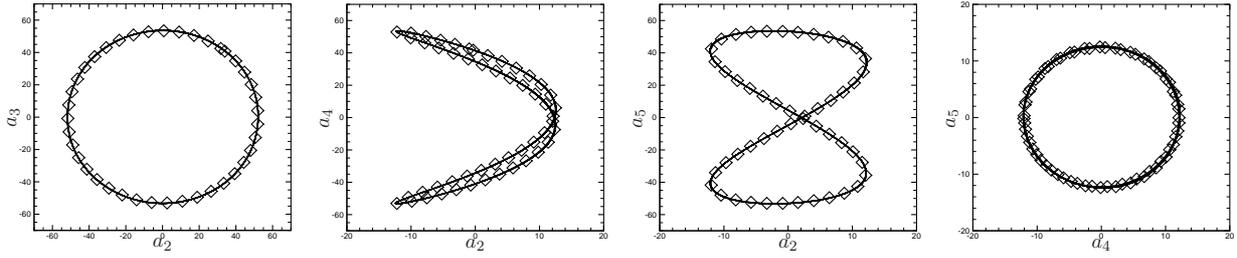


FIGURE 1.7 – Comparaison des cycles limites pour les modes projetés (NS : \diamond) et les modes prédits (ROM : —) sur 1000 périodes de lâché tourbillonnaire, avec 11 modes.

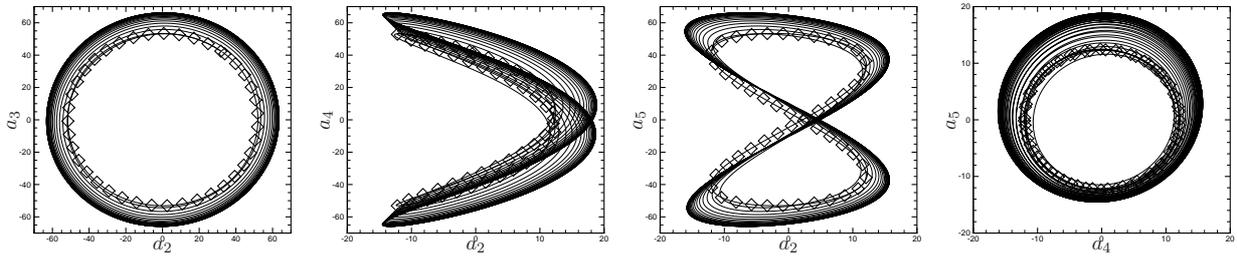


FIGURE 1.8 – Comparaison des cycles limites pour les modes projetés (NS : \diamond) et les modes prédits (ROM : —) sur 1000 périodes de lâché tourbillonnaire, avec 5 modes.

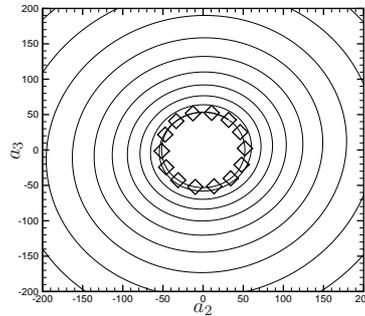


FIGURE 1.9 – Comparaison des cycles limites pour les modes projetés (NS : \diamond) et les modes prédits (ROM : —) sur 1000 périodes de lâché tourbillonnaire, avec 3 modes.

1.2 Amélioration de la stabilité du modèle réduit

1.2.1 Synthèse des méthodes de stabilisation existantes

Viscosité artificielle

La première classe de méthode est basée sur l'introduction des viscosités artificielles (Aubry *et al.*, 1988). La façon la plus simple consiste à ajouter une viscosité constante qui agit de la même manière sur tous les modes. Le modèle ainsi obtenu est appelé modèle d'Heisenberg (Podvin et Lumley, 1998; Delville *et al.*, 1999). La viscosité sans dimension $1/Re$ est alors remplacée par $(1+c)/Re$, et le problème consiste à déterminer la constante $c > 0$ pour obtenir un modèle réduit stable et précis. Rempfer et Fasel (1994) et Rempfer (1996) ont amélioré cette idée en supposant que la dissipation n'est pas identique pour chaque mode POD. La viscosité globale est alors remplacée par des viscosités différentes $1/Re_i = (1+c_i)/Re$ sur chaque modes POD Φ_i . Il est alors nécessaire de déterminer le jeu de coefficients c_i pour $i = 1, \dots, N_r$. Rempfer (1996) propose de définir

des viscosités fonction de l'index des couples de modes $j = 1, \dots, N_r/2$. Les coefficients c_j sont $c_j = K \times j$ où K est une constante à déterminer. Plus récemment, Karamanos et Karniadakis (2000) propose un modèle dissipative SVVM (Spectral Vanishing Viscosity Model). Le modèle est alors plus précis Sirisup et Karniadakis (2004). Dans cet esprit, nous avons proposé un modèle SVVM optimal basé sur l'adjoint (Bergmann *et al.*, 2005).

Calibration

La seconde classe de méthodes consiste à calibrer les coefficients polynomiaux du modèle réduit (Galletti *et al.*, 2004; Buffoni *et al.*, 2006; Couplet *et al.*, 2005; Kalb et Deane, 2007). Le tenseur B est déterminé par moindres carrés ou par une méthode adjointe pour minimiser l'erreur entre les coefficients prédits $a_i(t)$ et les vecteur propres temporels du tenseur de corrélations (equation (1.2)). Sur le fond, ces méthodes sont assez proches des méthodes SVVM.

Pénalisation

La troisième classe de méthodes est basée sur la pénalisation en ajoutant un terme supplémentaire dans le modèle. Cazemier (1997) et Cazemier *et al.* (1998) utilisent un opérateur cinétique modal pour déterminer les viscosités à ajouter sur chaque mode. Cazemier (1997) suppose que le manque d'interaction entre échelles résolues et non résolues est à l'origine de l'éventuelle divergence des modes temporels POD. Pour résoudre ce problème, un autre coefficient linéaire est introduit dans le système POD ROM qui devient :

$$\frac{da_i}{dt} = \mathcal{A}_i + \sum_{j=1}^{N_r} \mathcal{B}_{ij} a_j + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} \mathcal{C}_{ijk} a_j a_k + \mathcal{H}_i a_i,$$

où, après quelques manipulations basées sur des conservations énergétiques (Noack *et al.*, 2008) :

$$\mathcal{H}_i = -\frac{1}{\lambda_i} \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} \mathcal{C}_{ijk} \langle a_i a_j a_k \rangle - \mathcal{B}_{ii}.$$

Pour des écoulements compressibles, Vigo (1998) propose une stabilisation par ajout d'un terme cubique dans le modèle afin d'éviter une amplification non linéaire. Cette méthode semble donner de bons résultats mais la construction et la détermination des coefficients sont fastidieuses. Un terme linéaire peut être ajouté pour modéliser la pression. L'inconvénient relatif à toutes ces méthodes reste le coût CPU.

Stabilisation basée sur le résidu

Finalement, la dernière classe de méthode dans la littérature introduit une dissipation directement dans le schéma numérique utilisé pour construire le POD ROM (Iollo *et al.*, 2000a). A la place du produit scalaire standard L^2 , la norme de Sobolev H^1 est utilisée Iollo *et al.* (2000b). Cependant, le montant de dissipation doit toujours être fixé.

L'inconvénient majeur de toutes ces méthodes de stabilisation est qu'il y a toujours de nombreux paramètres à fixer ou à optimiser. Le but de la section suivante est de proposer des méthodes de stabilisation faisant intervenir le moins d'empirisme possible.

Soit $A^{[N_r]}$ le modèle défini par (1.20) avec N_r modes. On rappelle que des modèles instables sont obtenus pour $N_r = 3$ ou $N_r = 5$.

Dans ce qui suit nous allons présenter deux types de méthodes basées sur le résidu de l'opérateur de Navier-Stokes évalué avec les champs construits par POD. Ces résidus, appelés résidus POD-NS, sont :

$$\mathbf{R}_M(\mathbf{x}, t) = \frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\tilde{\mathbf{u}} \cdot \nabla) \tilde{\mathbf{u}} + \nabla \tilde{p} - \frac{1}{Re} \Delta \tilde{\mathbf{u}}, \quad (1.21a)$$

$$\mathbf{R}_C(\mathbf{x}, t) = \nabla \cdot \tilde{\mathbf{u}}, \quad (1.21b)$$

où les champs POD $\tilde{\mathbf{u}}$ et \tilde{p} sont donnés par (1.11).

Nous avons montré que le POD ROM est stable si un nombre suffisant de modes est conservé. Par exemple, le modèle $A^{[11]}$ est stable. Le modèle est stable si les champs POD sont proches des champs exacts (DNS). Les champs exacts s'écrivent :

$$\mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, t) + \mathbf{u}'(\mathbf{x}, t), \quad (1.22a)$$

$$p(\mathbf{x}, t) = \tilde{p}(\mathbf{x}, t) + p'(\mathbf{x}, t), \quad (1.22b)$$

où u' et p' sont les échelles non résolues par le POD ROM. Le calcul de ces petites échelles nécessite des coûts de calculs similaires à ceux générés pour le calculer les champs DNS. L'objectif est alors de dériver des méthodes de stabilisation qui utilisent des approximations de ces petites échelles non résolues.

1.2.2 Méthodes de stabilisation basées sur le résidu : Modèle $B^{[N_r;K]}$

Le but de cette partie est d'approximer les petites échelles u' et p' sur des fonctions de bases adaptées. Les champs exacts sont alors approximés par :

$$\mathbf{u}(\mathbf{x}, t) = \underbrace{\sum_{i=1}^{N_r} a_i(t) \phi_i(\mathbf{x})}_{\tilde{\mathbf{u}}(\mathbf{x}, t)} + \underbrace{\sum_{i=N_r+1}^{N_r+K} a_i(t) \phi'_i(\mathbf{x})}_{\mathbf{u}'(\mathbf{x}, t)}, \quad (1.23a)$$

$$p(\mathbf{x}, t) = \underbrace{\sum_{i=1}^{N_r} a_i(t) \psi_i(\mathbf{x})}_{\tilde{p}(\mathbf{x}, t)} + \underbrace{\sum_{i=N_r+1}^{N_r+K} a_i(t) \psi'_i(\mathbf{x})}_{p'(\mathbf{x}, t)}. \quad (1.23b)$$

Si les fonctions de base ϕ'_i et ψ'_i sont respectivement égales à ϕ_i et ψ_i l'optimalité énergétique est améliorée. Cependant, il n'est pas garanti que ces modes soient également optimaux pour stabiliser le modèle réduit. Par exemple, le modèle $A^{[5]}$, *i.e.* le modèle $B^{[3;2]}$ avec $\Phi' \equiv \Phi$ dans (1.23), n'est pas stable. Cependant, nous allons montrer qu'un autre modèle à 5 modes avec $N_r = 3$ et $K = 2$ est stable en utilisant d'autres fonctions Φ' . Il est bien connu que le résidu des équations régissant le problème joue un rôle majeur dans la stabilité des systèmes dynamiques (Bazilevs *et al.*, 2007). L'idée de cette section est de choisir ϕ'_i et ψ'_i comme étant la base POD des résidus POD-NS définis par (1.21). La méthode est la suivante.

Algorithme 1 (Stabilisation basée sur le résidu)

1. Intégrer le ROM $A^{[N_r]}$ pour obtenir $a_i(t)$ et calculer les N_s coefficients $a_i(t_k)$, $k = 1, \dots, N_s$.
2. Calculer les champs POD $\tilde{\mathbf{u}}(\mathbf{x}, t_k) = \sum_{i=1}^{N_r} a_i(t_k) \phi_i(\mathbf{x})$, $\tilde{p}(\mathbf{x}, t_k) = \sum_{i=1}^{N_r} a_i(t_k) \psi_i(\mathbf{x})$, et les résidus POD-NS $\mathbf{R}_M(\mathbf{x}, t_k)$ et $R_C(\mathbf{x}, t_k)$.
3. Calculer les modes POD $\phi'_i(\mathbf{x})$ et $\psi'_i(\mathbf{x})$ des résidus POD-NS $\mathbf{R}_M(\mathbf{x}, t_k)$ et $R_C(\mathbf{x}, t_k)$.
4. Ajouter les K premiers modes ϕ'_i et ψ'_i à la base existante ϕ_i et ψ_i (à l'aide d'une procédure de Gram-Schmidt) et construire le nouveau ROM (les conservation de la masse et du débit sont ici très importants).

Le modèle réduit obtenu avec cet algorithme est noté $B^{[N_r;K]}$ avec N_r le nombre de fonctions de base POD et K le nombre de modes POD des résidus. Les résultats de cette approche sont donnés à la section 1.2.4.

1.2.3 Méthodes SUPG et VMS : modèles $C^{[N_r]}$ et $D^{[N_r]}$

La méthode SUPG (Streamline Upwind Petrov-Galerkin) et la méthode VMS (Variational Multiscale) ont été développées pour la stabilisation des solutions numériques des équations Navier-Stokes. La méthode SUPG est une version simplifiée de la méthode VMS, et les principales étapes donnant ces modèles sont décrites dans Bazilevs *et al.* (2007).

L'idée de base des méthodes SUPG and VMS est d'approximer les petites échelles non résolues par :

$$\mathbf{u}' \simeq -\tau_M \mathbf{R}_M \quad (1.24a)$$

$$p' \simeq -\tau_C R_C, \quad (1.24b)$$

où τ_M et τ_C sont des constantes à fixer.

Les modèles réduits SUPG et VMS s'écrivent formellement :

$$\sum_{j=1}^{N_r} \left(L_{ij}^{(m)} + \beta L_{ij}^{(r)} \right) \frac{da_j}{dt} = \sum_{j=1}^{N_r} \left(B_{ij}^{(m)} + \alpha B_{ij}^{(c)} \right) a_j + \sum_{j=1}^{N_r} \sum_{k=1}^{N_r} C_{ijk}^{(m)} a_j a_k + F_i(t), \quad (1.25)$$

où le terme de pénalisation $F_i(t)$ est défini par :

– Modèle SUPG $C^{[N_r]}$:

$$F_i^{SUPG}(t) = (\tilde{\mathbf{u}} \cdot \nabla \phi_i + \nabla \psi_i, \tau_M \mathbf{R}_M(\mathbf{x}, t))_\Omega + (\nabla \cdot \phi_i, \tau_C R_C(\mathbf{x}, t))_\Omega. \quad (1.26)$$

– Modèle VMS $D^{[N_r]}$:

$$F_i^{VMS}(t) = F_i^{SUPG}(t) + (\tilde{\mathbf{u}} \cdot (\nabla \phi_i)^T, \tau_M \mathbf{R}_M(\mathbf{x}, t))_\Omega - (\nabla \phi_i, \tau_M \mathbf{R}_M(\mathbf{x}, t) \otimes \tau_M \mathbf{R}_M(\mathbf{x}, t))_\Omega \quad (1.27)$$

Les paramètres τ_M et τ_C peuvent être déterminés par des considérations d'échelles (Hugues *et al.*, 1998), et aucune modélisation ou empirisme n'est alors nécessaire. Ceci donne un modèle de fermeture POD ROM "universel". Dans ce qui suit les paramètres τ_M et τ_C seront déterminés par une procédure optimale de telle sorte que les coefficients temporels $a_i(t)$ soient aussi proches que possible des vecteurs propres temporels du tenseur des corrélations (1.3).

La table 1.1 synthétise les différents modèles, où N_r est le nombre de modes POD utilisés pour construire le modèle réduit POD ROM.

ROM	Method
$A^{[N_r]}$	Sans stabilisation
$B^{[N_r, K]}$	Stabilisation avec K modes POD de résidus
$C^{[N_r]}$	Stabilisation SUPG
$D^{[N_r]}$	Stabilisation VMS

TABLE 1.1 – Description des différents modèles réduits. N_r est le nombre de modes POD utilisés pour construire le modèle réduit POD .

1.2.4 Résultats numériques des méthodes de stabilisation

La configuration d'écoulement et les paramètres utilisés sont les mêmes que ceux introduits à la section §1.1.3. Le nombre de Reynolds est $Re = 200$ et nous avons utilisé $N_s = 80$ snapshots uniformément répartis sur une période tourbillonnaire. Nous étudions les configurations instables observées à la section §1.1.3, *i.e.* $N_r = 5$ et $N_r = 3$. Pour le modèle $B^{[N_r; K]}$ nous utilisons $K = 2$ modes de résidus. Ces deux modes sont très différents des modes POD 4 et 5 utilisés pour $A^{[5]}$ (figure 1.10).

Les cycles limites obtenus avec les modèles B , C et D sont représentés sur les figures 1.11 et 1.12, respectivement pour $N_r = 5$ et $N_r = 3$. Ces cycles limites représentent 1000 périodes. Ces cycles limites sont comparés aux cycles limites obtenus par DNS, et nous observons un très bon accord entre tous les cycles limites.

Afin de mettre en évidence les différences entre chaque modèle nous étudions la norme L^2 des résidus POD-NS (1.21). L'évolution temporelle de la norme L^2 des résidus POD-NS obtenus par le modèle A (modèle (1.20) ou (1.25) avec $F_i = 0$) avec $N_r = 5$ et $N_r = 3$ est représentée sur les figures 1.13(a) et 1.14(a), respectivement. Nous observons une croissance initiale avant d'atteindre un régime asymptotique. Ceci peut être expliqué par le fait que la dynamique converge vers un attracteur erroné (figure 1.7). Nous observons même une divergence pour $N_r = 3$ (en accord avec la figure 1.9). Nous comparons l'efficacité des modèles B , C et D sur les figures 1.13(b) et 1.14(b). Tous les modèles stabilisés donnent des solutions précises (norme des résidus POD-NS faible). Pour $N_r = 5$ ou $N_r = 3$, les modèles C et D sont plus précis que le modèle B . Comme on pouvait s'y attendre, le modèle VMS est même un petit plus précis que le modèle SUPG. En conclusion pour la première méthode de stabilisation (modèle B), nous observons que le modèle à 5 modes $B^{[3; 2]}$ est stable et précis alors que le modèle à 5 modes $A^{[5]}$ ne l'est pas. La stratégie qui consiste à ajouter beaucoup de modes POD pour stabiliser un modèle réduit n'est pas optimale. Il vaut mieux en effet ajouter des modes appropriés comme les modes d'amortissement (figure 1.10).

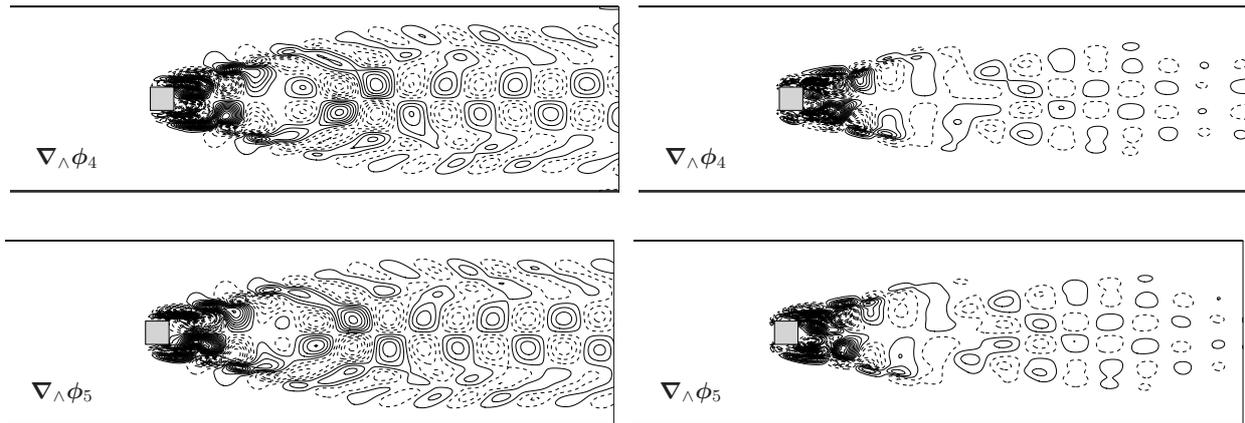


FIGURE 1.10 – Comparaison des modes POD originaux (gauche, modèle $A^{[5]}$) avec les modes de résidus (droite, modèle $B^{[3;2]}$). Les valeurs négatives sont représentées en pointillés.

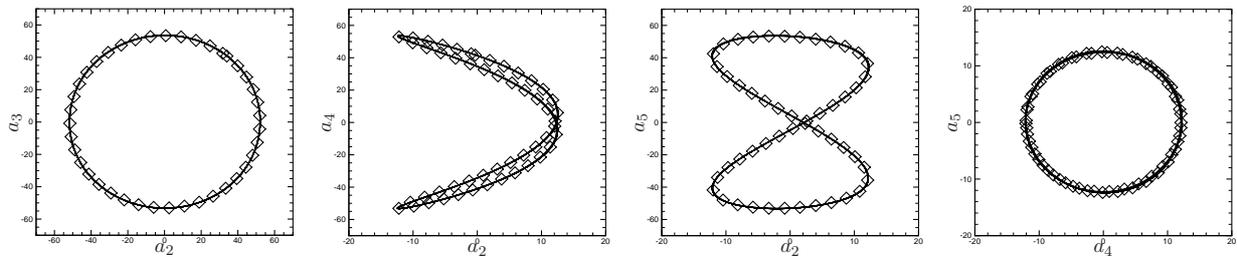


FIGURE 1.11 – Comparaison des cycles limites DNS projetés (NS : \diamond) et prédits (ROM : —) sur 1000 périodes pour les modèles $B^{[5,2]}$, $C^{[5]}$ et $D^{[5]}$ (tous les cycles sont confondus).

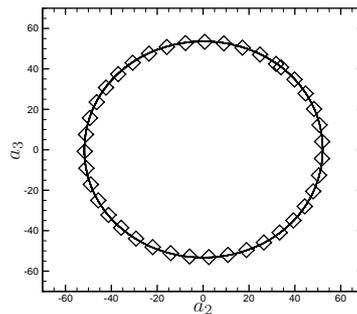


FIGURE 1.12 – Comparaison des cycles limites DNS projetés (NS : \diamond) et prédits (ROM : —) sur 1000 périodes pour les modèles $B^{[3,2]}$, $C^{[3]}$ et $D^{[3]}$ (tous les cycles sont confondus).

1.3 Amélioration de la robustesse de la base POD

La POD a été initialement introduite par Lumley (1967) en 1967 pour identifier les structures cohérentes d'un écoulement turbulent. Plus récemment, les modèles réduits POD ROM ont été utilisés pour des problèmes de contrôle d'écoulement (Ito et Ravindran, 1998; Ravindran, 2000; Graham *et al.*, 1999b; Bergmann *et al.*, 2005; Bergmann et Cordier, 2008). Ceci permet de réduire significativement les coûts de calcul CPU, notamment pour des optimisations par des méthodes adjointes. Le principal inconvénient pour le contrôle est

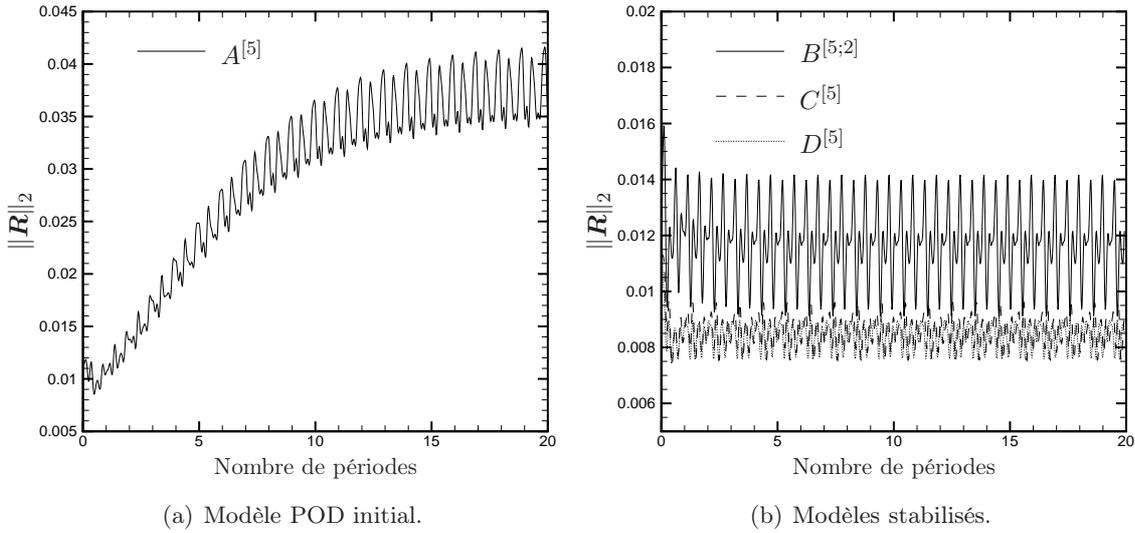


FIGURE 1.13 – Evolution temporelle de la norme L^2 des résidus POD-NS avec 5-modes.

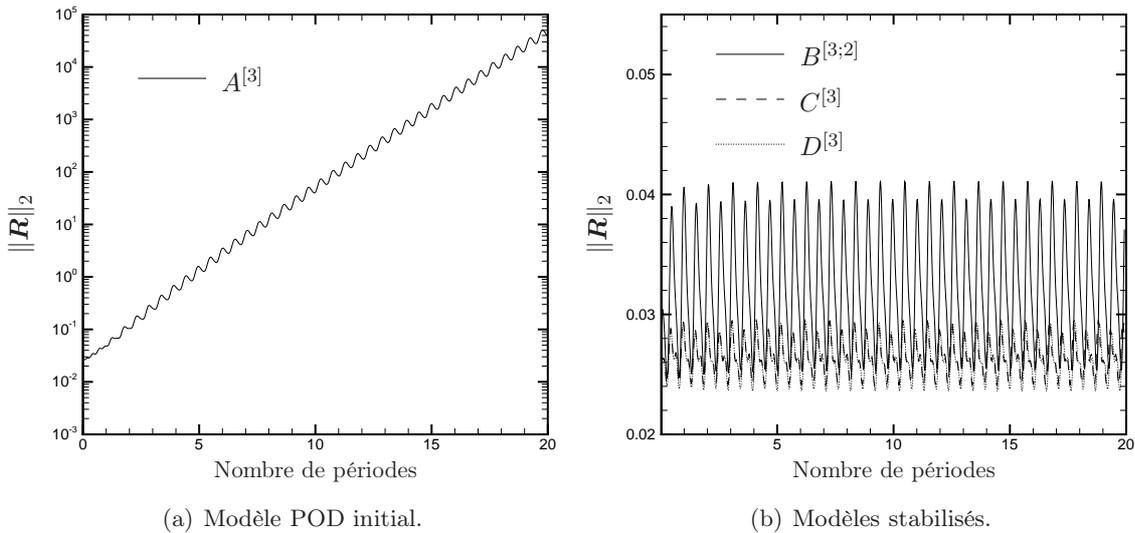


FIGURE 1.14 – Evolution temporelle de la norme L^2 des résidus POD-NS avec 3-modes.

que la base POD est uniquement capable de donner une représentation optimale des snapshots utilisés pour construire la base (Prabhu *et al.*, 2001; Noack *et al.*, 2003; Bergmann et Cordier, 2008). Pour le contrôle d'écoulement, différentes stratégies peuvent être utilisées pour construire des fonctions de base adaptées.

La première méthode, que nous décrivons à la fin de ce chapitre, consiste à générer une base globale qui est robuste pour différents paramètres d'entrée du système. Par exemple il est possible de générer des snapshots pour différentes lois de contrôle (Bergmann et Cordier, 2008) ou différents nombres de Reynolds (Ma et Karniadakis, 2002). Une solution efficace que nous détaillerons est basée sur la CVT (*Centroidal Voronoi Tessellations*), initialement proposée par Burkardt *et al.* (2004). Une autre solution que nous avons également étudiée est de générer une large gamme de dynamiques en utilisant une loi de contrôle *ad hoc* (Bergmann *et al.*, 2005).

La seconde méthode est d'actualiser la base POD pendant le processus d'optimisation. La méthode que nous avons développée, *Trust Region Proper Orthogonal Decomposition* (TRPOD), basée sur des idées de Fahl (2000), illustre bien cette idée (Bergmann et Cordier, 2008). Le principal défaut de la TRPOD est que la base POD doit être entièrement réactualisée par DNS et nécessite de ce fait des coûts de calcul qui peuvent être importants. Dans ce qui suit nous allons présenter des méthodes efficaces pour actualiser la base POD.

Par exemple, nous allons essayer de construire une base POD pour une nombre de Reynolds $Re_2 = 200$ en partant d'une base POD construite pour $Re_1 = 100$. Ces bases POD sont relativement différentes comme le montre la figure 1.15.

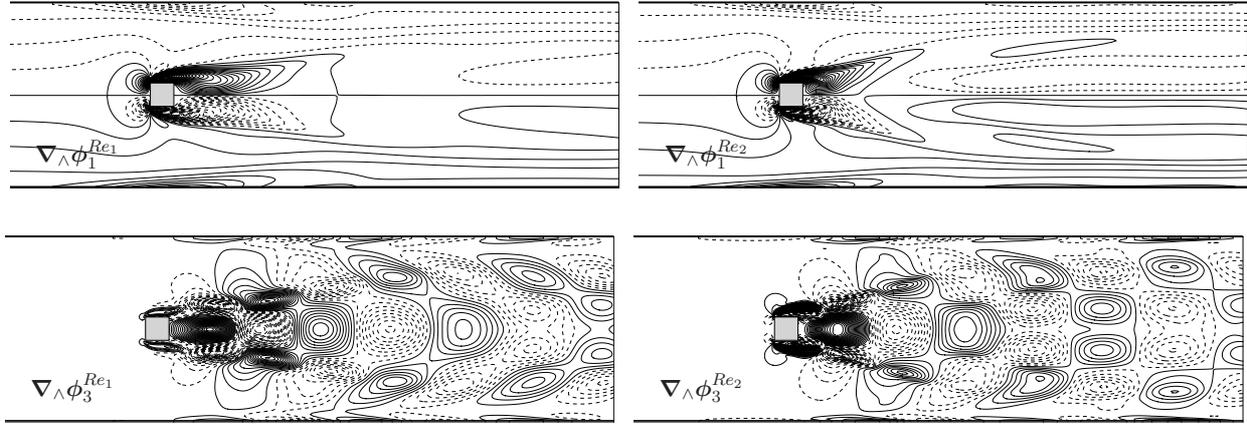


FIGURE 1.15 – Comparaison de quelques modes POD en iso-vorticité pour $Re_1 = 100$ (gauche) et $Re_2 = 200$ (droite).

Nous allons présenter deux méthodes. La première méthode consiste à améliorer la représentativité de la base en utilisant des approximations résiduelles de l'information manquante \mathbf{u}' and p' . Il s'agit d'une mise à jour itérative (§1.3.1) qui utilise la même idée que celle présentée à la section §1.2.2. La seconde méthode utilise des simulations DNS pour mettre à jour la base POD. Ces deux méthodes sont itératives. Notons $\Phi \equiv \Phi^{(n)}$ la base obtenue au cours du processus itératif avec n le nombre d'itérations effectuées.

Nous avons $\Phi^{(0)} = \Phi^{Re_1}$ et espérons $\Phi^{(+\infty)} = \Phi^{Re_2}$. Le critère de convergence utilisé ici est la meilleure projection de $\Phi^{(n)}$ sur Φ^{Re_2} , notée $\Phi \cdot \Phi^{Re_2}$. Dans notre cas, pour $Re_1 = 100$ et $Re_2 = 200$, la valeur initiale est $\Phi^{Re_1} \cdot \Phi^{Re_2} \approx 0.5$. La valeur désirée serait $\Phi^{(n)} \cdot \Phi^{Re_2} \approx 1$, avec le nombre d'itération n le plus petit possible.

1.3.1 Une méthode basée sur les sous espaces de Krylov

Cette méthode est basée sur une application successive de l'opérateur de Navier-Stokes sur les résidus. Il s'agit en fait de la version itérative de l'algorithme de stabilisation présenté à la section §1.2.2. L'adaptation du sous espace POD est décrit en détail par l'algorithme 2 et est schématiquement représenté sur la figure 1.16. L'intégration du POD ROM (1.20) est toujours effectuée avec le nombre de Reynolds $Re \equiv Re_2$ (le nombre de Reynolds Re intervient dans l'évaluation du terme $B^{(m)}$, voir équation (1.15b)). Tous les autres coefficients, i.e. $L^{(m)}$, $L^{(r)}$, $B^{(m)}$, $B^{(c)}$, $C^{(m)}$, sont construits à partir la base POD courante, $\Phi = \Phi^{(n)}$, suivant (1.15).

Algorithme 2 (Adaptation de la base POD par une méthode de Krylov)

Commencer avec la base POD à adapter, Φ_i avec $i = 1, \dots, N_r$ (dans notre cas construite avec $Re = Re_1$). Soit $N_0 = N_r$ et $\mathcal{T} = [0, T]$ une période d'observation.

1. *Construire et résoudre le modèle réduit ROM correspondant sur \mathcal{T} avec $Re = Re_2$ pour obtenir $a_i(t)$ et extraire N_s snapshots $a_i(t_k)$ avec $i = 1, \dots, N_r$ et $k = 1, \dots, N_s$. Calculer les champs $\tilde{\mathbf{u}}(\mathbf{x}, t_k)$, $\tilde{p}(\mathbf{x}, t_k)$ à partir de l'équation (1.11).*
2. *Calculer les résidus POD-NS $\mathbf{R}(\mathbf{x}, t_k) = (\mathbf{R}_M(\mathbf{x}, t_k), R_C(\mathbf{x}, t_k))^T$ à partir de l'équation (1.21).*
3. *Calculer les modes POD $\tilde{\Phi}(\mathbf{x}) = (\tilde{\phi}(\mathbf{x}), \tilde{\psi}(\mathbf{x}))^T$ à partir de la base de données des résidus POD-NS $\mathbf{R}(\mathbf{x}, t_k)$, $k = 1, \dots, N_s$.*
4. *Ajouter les K premiers modes POD de résidus $\tilde{\Phi}(\mathbf{x})$ à la base POD existante $\Phi_i(\mathbf{x})$ (avec une procédure de Gram Schmidt)*
 - $\Phi \leftarrow \Phi + \Psi$

- $N_r \leftarrow N_r + K$
 - Si N_r est inférieur à un seuil N_{max} retourner à l'étape 1. Sinon, aller à 5.
5. Faire l'étape (1). A partir des champs $\tilde{\mathbf{u}}$ et \tilde{p} , effectuer une nouvelle compression POD avec $N_r = N_0$.
- Si un critère de convergence est satisfait, arrêter. Sinon, retourner à l'étape 1.

Cet algorithme est une version simplifiée de l'algorithme GMRES (generalized minimal residual) pour les systèmes linéaires (Saad et Schultz, 1986). Avant de tester cette méthode sur les équations de Navier-Stokes

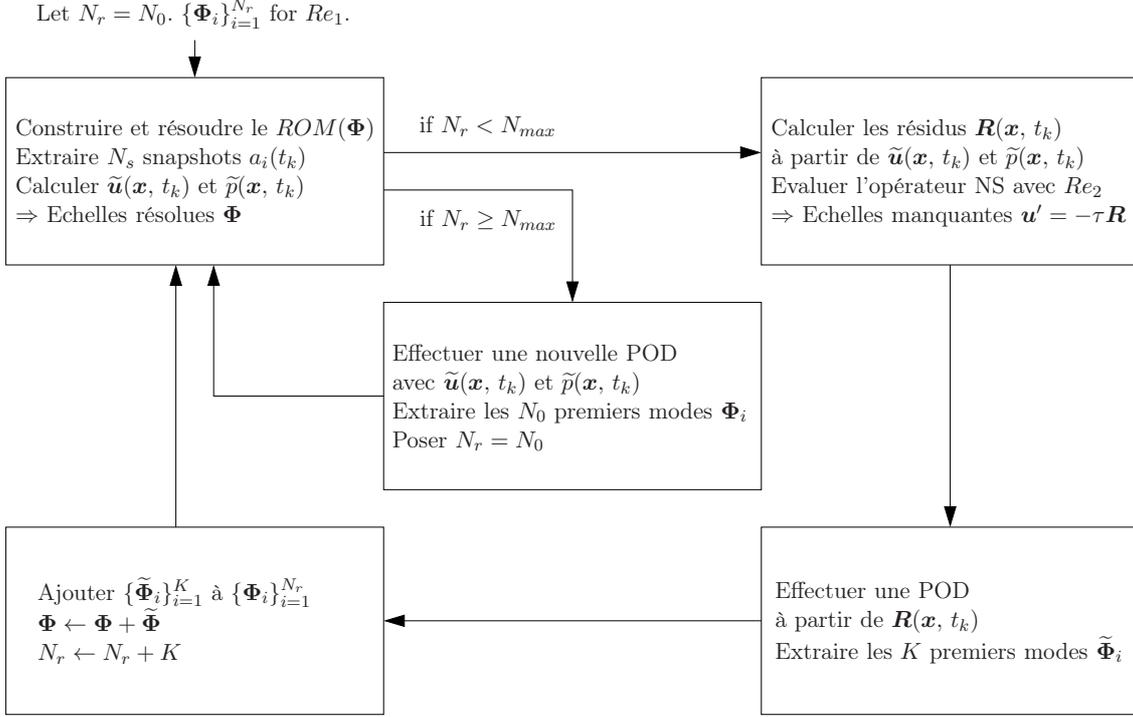


FIGURE 1.16 – Représentation schématique de l'adaptation de la base POD avec une méthode de type Krylov.

(§1.3.1), nous allons la tester sur un modèle simple de l'équation de Burgers 1D (§1.3.1).

Equation de Burgers 1D

L'équation de Burgers dans sa forme sans dimension est :

$$\mathcal{L}_B(u) = \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} - \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} = 0, \quad (1.28)$$

avec une condition initiale ⁵

$$u(x, 0) = \sin\left(\pi \frac{\tan(c_s(2x-1))}{\tan(c_s)}\right) \quad \text{and} \quad c_s = 1.3, \quad (1.29)$$

et les conditions aux limites

$$\begin{aligned} u(0, t) &= 0, \\ u(L, t) &= 0. \end{aligned} \quad (1.30)$$

Cette équation est résolue dans le domaine \mathcal{D} défini par :

$$\mathcal{D} = \{(x, t) \in [0, 1] \times [0, 1]\}.$$

5. La valeur de c_s est choisie pour favoriser un choc dans le domaine de calcul \mathcal{D} .

Nous choisissons une base POD initiale Φ^{Re_1} et une base POD cible Φ^{Re_2} de telle sorte que $\Phi^{Re_1} \cdot \Phi^{Re_2} \approx 0.5$, *i.e.* $Re_1 = 50$ et $Re_2 = 300$ dans ce contexte. Nous utilisons 40 snapshots uniformément répartis sur $\mathcal{T} = [0, 1]$. Une comparaison de quelques modes POD pour Re_1 et Re_2 est représentée sur la figure 1.17. Des différences significatives peuvent être observées. L'évolution du critère de convergence en fonction de nombre d'itération

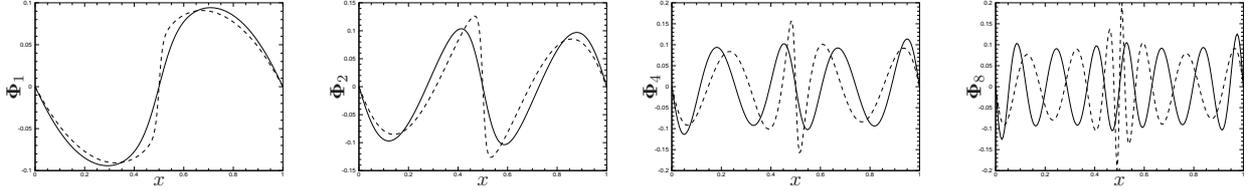


FIGURE 1.17 – Comparaison de quelques modes POD pour $Re_1 = 50$ (—) et $Re_2 = 300$ (---).

est présentée sur 1.18. La convergence est obtenue avec environ 6 itérations. Nous avons donc adapté la base POD sans DNS. La base POD pour Re_2 peut même être déterminée en partant d'un seul mode (la condition initiale normalisée par exemple), ou alors en partant de n'importe quelle autre base. Il s'agit donc d'une méthode très efficace pour l'équation de Burgers. Voyons maintenant si cette méthode est aussi performante pour les équations de Navier-Stokes.

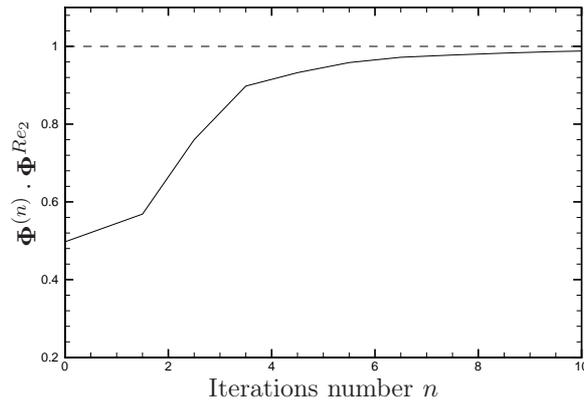


FIGURE 1.18 – Evolution du critère de convergence en fonction du nombre d'itérations pour l'équation de Burgers 1D.

Equations de Navier-Stokes 2D pour le cylindre confiné

Le cas test est l'écoulement confiné autour d'un cylindre de section carré en régime laminaire. Nous prenons $Re_1 = 100$ et $Re_2 = 200$, et nous extrayons 40 snapshots uniformément distribués sur une période tourbillonnaire T , et donc $\mathcal{T} = [0, T]$. Naturellement, au cours du processus d'adaptation de la base nous devons également adapter la période T . Comme le montre la figure 1.19 nous n'obtenons pas de convergence. L'algorithme est arrêté lorsque les coûts de calcul dépassent les coûts DNS (l'algorithme ne serait plus performant). Des résultats similaires sont obtenus en utilisant 400 snapshots sur $\mathcal{T} = [0, 10T]$. L'information contenue dans les résidus de Navier-Stokes n'est pas suffisante pour adapter la base POD, mais ils sont tout de même nécessaires pour stabiliser le modèle réduit pour une dynamique donnée (voir §1.2.2). Une explication possible est que l'approximation de échelles manquantes, $\mathbf{u}'(\mathbf{x}, t) = -\tau_M \mathbf{R}_M(\mathbf{x}, t)$ et $p'(\mathbf{x}, t) = -\tau_C R_C(\mathbf{x}, t)$, est uniquement précise pour des très petites échelles, *i.e.* celles non retenues par la troncature POD. Les modes de résidus ont donc uniquement un impact dissipatif. On peut alors se poser la question de savoir s'il est possible de déterminer des paramètres τ_M et τ_C pour des échelles manquantes qui ne sont pas forcément petites. La réponse n'est pas évidente, et une solution serait peut être de regarder $\mathbf{U}'(\mathbf{x}, t) = M(t)\mathbf{R}(\mathbf{x}, t)$, avec $M \in \mathbb{R}^{3 \times 3}$ pour les équations de Navier-Stokes 2D. Un pas en direction de l'algorithme GMRES complet a été effectué. Quelques modes d'Arnoldi, $\Phi_n'' = A\Phi_n'$, $\Phi_n''' = A\Phi_n''$, ..., ont été ajoutés à la base initiale.

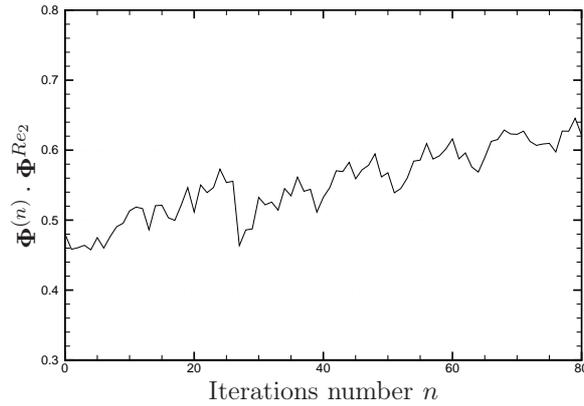


FIGURE 1.19 – Evolution de critère de convergence en fonction du nombre d’itérations pour les équations de Navier-Stokes 2D.

L’opérateur A représente l’opérateur linéaire obtenu une discrétisation adaptée des équations de Navier-Stokes, et donc $A\mathbf{U} = b(\mathbf{U})$. Les résultats n’ont pas réellement été améliorés. Une base d’Arnold relativement large serait nécessaire et les coûts de calcul rendraient la méthode inefficace.

La section suivante présente un autre type d’algorithme qui couple POD ROM et DNS. Si la solution des équations de Navier-Stokes vit sur un attracteur bien défini (en particulier, pas de changement de dynamique), la DNS pourra être grandement accélérée en utilisant des fonctions de base POD (Sirisup *et al.*, 2005) .

1.3.2 Une méthode hybride DNS/POD ROM

Il a été démontré que la représentativité énergétique POD décroît rapidement en dehors de l’intervalle temporel de la base de données (snapshots) pour des écoulements 3D (Buffoni *et al.*, 2006). Le but de ce qui suit est alors de mettre à jour la base de données quand le temps évolue. De cette façon la base POD est actualisée et représente fidèlement l’écoulement. L’idée est de remplacer les plus anciens snapshots par de nouveaux, si possible avec un coût CPU réduit. Ces nouveaux snapshots sont obtenus en utilisant quelques simulations du modèle détaillé (DNS). A partir de cette nouvelle base, un nouveau modèle réduit est construit et intégré jusqu’à qu’un nouveau snapshot soit nécessaire. Cette procédure est alors répétée. La figure 1.20 présente schématiquement l’algorithme utilisé. Posons Re_1 le nombre de Reynolds utilisé pour construire la base POD initiale, et Re_2 le nombre de Reynolds associé à la nouvelle dynamique que l’on souhaite simuler. Toutes les simulations (ROM et DNS sur la figure 1.20) sont effectuées avec $Re = Re_2$. Après quelques itérations DNS,

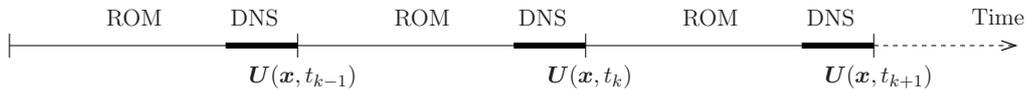


FIGURE 1.20 – Représentation schématique d’une méthode hybride DNS-POD ROM pour améliorer la base POD.

un nouveau snapshot est disponible. Avec ce snapshot nous actualisons la base de données (§1.3.2), puis nous calculons le tenseur de corrélations (§1.3.2) et enfin nous calculons la nouvelle base POD (§1.3.2). Tous les coefficients du ROM (1.20), *i.e.* $L^{(m)}$, $L^{(r)}$, $B^{(m)}$, $B^{(e)}$, $C^{(m)}$, sont construits avec la base POD actualisée, et l’intégration est effectuée avec $Re \equiv Re_2$. En effet, le nombre de Reynolds Re est nécessaire pour calculer $B^{(m)}$ (voir (1.15b)). Pour obtenir une actualisation rapide et efficace de la base POD, des méthodes performantes sont développées pour chaque étapes mentionnées ci-dessus.

Modification de la base de données

Chaque snapshot de la base de donnée peut être évalué avec l'équation (1.7) :

$$\widehat{\mathbf{U}}^{[1,\dots,N_r]}(\mathbf{x}, t_k) = \sum_{n=1}^{N_r} a_n(t_k) \Phi_n(\mathbf{x}). \quad (1.31)$$

Le nouveau snapshot peut être ajouté à la base de donnée ou alors il remplace le snapshot le plus ancien (supposé le moins adapté à la nouvelle dynamique). La position du nouveau snapshot est notée s . Le nombre de snapshots inclus dans la base de données est alors $N = \max(s, N_r)$. Le snapshot \mathbf{U}_s peut être décomposé linéairement sur la base POD plus une partie orthogonale :

$$\widetilde{\mathbf{U}}(\mathbf{x}, t_s)^{[1,\dots,N_r]} = \widehat{\mathbf{U}}^{[1,\dots,N_r]}(\mathbf{x}, t_s) + \mathbf{U}_s^\perp(\mathbf{x}, t_s).$$

De manière générale on a :

$$\widetilde{\mathbf{U}}(\mathbf{x}, t_k)^{[1,\dots,N_r]} = \widehat{\mathbf{U}}^{[1,\dots,N_r]}(\mathbf{x}, t_k) + \delta_{ks} \mathbf{U}^\perp(\mathbf{x}, t_s),$$

où δ est le symbole de Kronecker.

Evaluation du tenseur des corrélations temporelles

Afin de réduire les coûts de calcul CPU le tenseur des corrélations C est calculé en utilisant la décomposition $\widetilde{\mathbf{U}}_k^{[1,\dots,N_r]}$.

$$\begin{aligned} C(t_k, t_l) &= \left(\widetilde{\mathbf{U}}(\mathbf{x}, t_k)^{[1,\dots,N_r]}, \widetilde{\mathbf{U}}(\mathbf{x}, t_l)^{[1,\dots,N_r]} \right)_\Omega \\ &= \left(\sum_{i=1}^{N_r} a_i(t_k) \phi_i(\mathbf{x}) + \delta_{ks} \mathbf{U}^\perp(\mathbf{x}, t_k), \sum_{j=1}^{N_r} a_j(t_l) \phi_j(\mathbf{x}) + \delta_{ls} \mathbf{U}^\perp(\mathbf{x}, t_l) \right)_\Omega \\ &= \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} a_i(t_k) a_j(t_l) \underbrace{(\phi_i(\mathbf{x}), \phi_j(\mathbf{x}))_\Omega}_{=\delta_{ij}} + \delta_{ks} \delta_{ls} (\mathbf{U}^\perp(\mathbf{x}, t_k), \mathbf{U}^\perp(\mathbf{x}, t_l))_\Omega \\ &\quad + \delta_{ls} \sum_{i=1}^{N_r} a_i(t_k) \underbrace{(\phi_i(\mathbf{x}), \mathbf{U}^\perp(\mathbf{x}, t_l))_\Omega}_{=0} + \delta_{ks} \sum_{j=1}^{N_r} a_j(t_l) \underbrace{(\mathbf{U}^{\perp T}(\mathbf{x}, t_k), \phi_j(\mathbf{x}))_\Omega}_{=0}. \end{aligned}$$

Ainsi, une approximation du tenseur des corrélations est :

$$C(t_k, t_l) = \sum_{i=1}^{N_r} a_i(t_k) a_i(t_l) + \delta_{ks} \delta_{ls} \int_\Omega \sum_{i=1}^{n_c} U^{\perp i}(\mathbf{x}, t_k) U^{\perp i}(\mathbf{x}, t_l) d\mathbf{x}. \quad (1.32)$$

L'évaluation de cette matrice est très rapide. En effet, il n'est pas nécessaire de calculer toutes les corrélations en espace mais uniquement sur les coefficients temporels plus l'auto-corrélation \mathbf{U}_s^\perp pour évaluer la composante C_{ss} .

Actualisation de la base POD

Toutes les quantités évaluées avant le calcul du tenseurs des corrélations (1.32) sont notées avec l'exposant (n) . La base POD actualisée peut être évaluée par (voir Sirovich (1987) pour plus de détails) :

$$\begin{aligned} \phi_k^{(n+1)}(\mathbf{x}) &= \frac{1}{\lambda_k^{(n+1)}} \sum_{j=1}^{N_r} \widetilde{\mathbf{U}}^{(n)}(\mathbf{x}, t_j) a_k^{(n+1)}(t_j) \\ &= \frac{1}{\lambda_k^{(n+1)}} \sum_{j=1}^{N_r} \left(\sum_{i=1}^{N_r} a_i^{(n)}(t_j) \phi_i^{(n)}(\mathbf{x}) + \delta_{js} \mathbf{U}^{\perp(n)}(\mathbf{x}, t_j) \right) a_k^{(n+1)}(t_j) \\ &= \frac{1}{\lambda_k^{(n+1)}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} a_k^{(n+1)}(t_j) a_i^{(n)}(t_j) \phi_i^{(n)}(\mathbf{x}) + \frac{1}{\lambda_k^{(n+1)}} \mathbf{U}^{\perp(n)}(\mathbf{x}, t_s) a_k^{(n+1)}(t_s). \end{aligned}$$

Soit $K^{(n+1)} \in \mathbb{R}^{N_s \times N_s}$ le tenseur des corrélations modales entre nouvelles et anciennes fonctions propres temporelles,

$$K_{ki}^{(n+1)} = \frac{1}{\lambda_k^{(n+1)}} \sum_{j=1}^{N_r} a_k^{(n+1)}(t_j) a_i^{(n)}(t_j),$$

et soit

$$\mathbf{S}_k^{(n+1)}(\mathbf{x}) = \frac{1}{\lambda_k^{(n+1)}} \mathbf{U}^\perp{}^{(n)}(\mathbf{x}, t_s) a_k^{(n+1)}(t_s),$$

Nous avons :

$$\phi_k^{(n+1)}(\mathbf{x}) = \sum_{i=1}^{N_r} K_{ki}^{(n+1)} \phi_i^{(n)}(\mathbf{x}) + \mathbf{S}_k^{(n+1)}(\mathbf{x}). \quad (1.33)$$

En utilisant la matrice $S^{(n+1)}$ avec les éléments $S_{ij}^{(n+1)} = S_i^j{}^{(n+1)}$, la nouvelle base POD peut être obtenue de l'ancienne à partir de l'application linéaire $\varphi : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n \times \mathbb{R}^n$ définie par :

$$\varphi : \phi^{(n)} \mapsto \phi^{(n+1)} = \phi^{(n)} K^{(n+1)} + S^{(n+1)}. \quad (1.34)$$

L'actualisation de la base POD est alors très rapide (il n'est pas nécessaire de reconstruire une nouvelle base).

Résultats numériques

Le tenseur des corrélations est évalué avec 40 snapshots uniformément répartis sur une période. Nous testons tout d'abord la méthode d'approximation du tenseur des corrélations introduit à §1.3.2. Une comparaison des valeurs propres entre tenseur exact (évalué avec \mathbf{U}) et approché avec $\tilde{\mathbf{U}}^{[1, \dots, N_r]}$ (avec $N_r = 5$ et $N_r = 11$ modes) est présentée sur la figure. (1.21). Dans les deux cas l'information relative $RIC(N_r)$ est supérieure à 99%. Les N_r premières valeurs propres approchées sont en très bon accord avec les valeurs propres exactes. Toutes les autres valeurs propres pour $n > N_r$ approchées sont égale à zéro sauf pour λ_{N_r+1} , ce qui indique une contribution non nulle de la partie orthogonale \mathbf{U}^\perp .

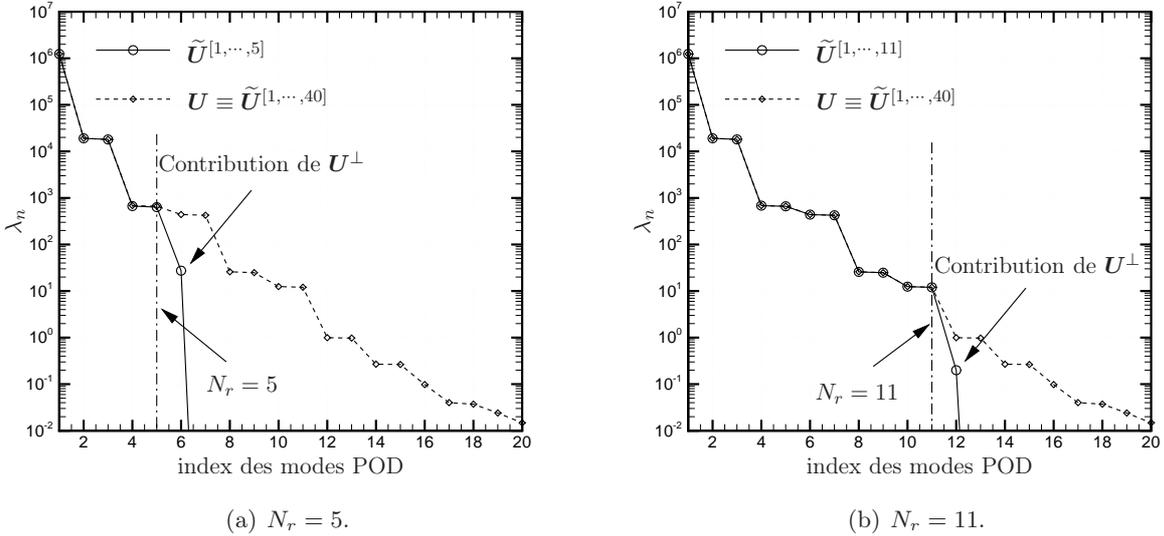


FIGURE 1.21 – Comparaison des valeurs propres entre tenseur exact (évalué avec \mathbf{U}) et approché avec $\tilde{\mathbf{U}}^{[1, \dots, N_r]}$ (avec $N_r = 5$ et $N_r = 11$ modes).

Les résultats de l'actualisation de la base POD avec la transformation linéaire introduit à la section 1.3.2 est présentée sur la figure 1.22. Une étape d'actualisation de l'écoulement transitoire entre $Re_1 = 100$ et $Re_2 = 200$ est présentée.

Pour la méthode hybride POD-DNS nous considérons la base initiale Φ^{Re_1} et la base cible Φ^{Re_2} , qui correspondent à des nombres de Reynolds $Re_1 = 100$ et $Re_2 = 200$. La figure 1.23 présente l'évolution

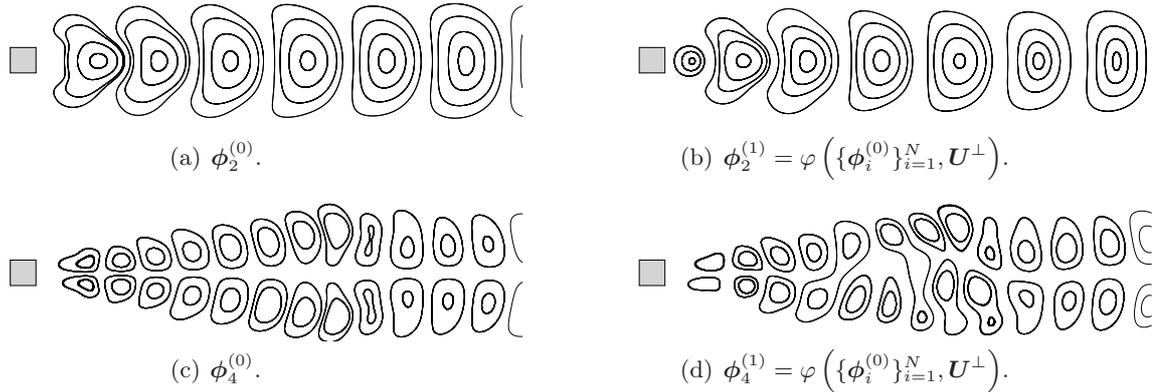


FIGURE 1.22 – Modification des fonctions de base POD avec l'application linéaire φ . Représentation en lignes de courant.

du critère de convergence $\Phi \cdot \Phi^{Re_2}$ en fonction du nombre de périodes tourbillonnaires (et donc le nombre d'actualisation de bases) pour différents pourcentages de DNS. On note T_{NS} et T_{ROM} les intervalles de temps utilisés par la DNS et le POD ROM⁶ et le pourcentage de DNS est défini par $P_{NS} = T_{NS}/(T_{NS} + T_{ROM})$.

La simulation avec $P_{NS} = 100\%$ ne fait pas intervenir de POD ROM. Seulement 10 périodes tourbillonnaires sont suffisantes pour converger vers la bonne base POD. Ce temps est nécessaire pour atteindre le régime asymptotique à Re_2 . Des résultats similaires sont obtenus avec $P_{NS} = 90\%$ et $P_{NS} = 80\%$, et également avec une précision moindre avec $P_{NS} = 80\%$. Cependant, nous n'observons pas de réelle convergence avec $P_{NS} = 70\%$. Un grand pourcentage de DNS est tout de même nécessaire pour atteindre la convergence. Une explication de ce phénomène est donné sur la figure 1.24. Puisque la base POD n'est pas bien adaptée pour donner une bonne représentation de l'écoulement courant la solution s'éloigne du bon cycle limite. Si la DNS est capable de corriger cette erreur (voir figures 1.24(a) and 1.24(b)), la convergence est obtenue. Sinon (voir figure 1.24(c)), la convergence ne peut pas être atteinte

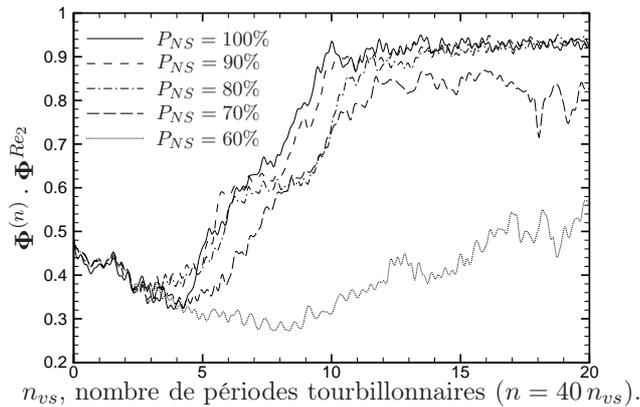


FIGURE 1.23 – Evolution de critère de convergence en fonction de nombre de période tourbillonnaires.

Le temps de calcul engendré par le ROM est négligeable par rapport au temps de calcul DNS. De même l'actualisation de la base décrite dans la section 1.3.2, 1.3.2 et 1.3.2 ne nécessitent pas de coûts de calcul importants. En pratique nous avons estimé une réduction globale des coûts de calcul de la méthode hybride POD-DNS d'environ 20% en comparaison avec une DNS pour un changement de dynamique (modification du Reynolds). Naturellement, sur une dynamique bien définie, nous pourrions utiliser presque 100% de ROM. Les POD ROM peuvent donc être considérés comme de vrais accélérateurs de simulations numériques.

6. Le POD ROM autorise des pas de temps plus grands que la DNS.

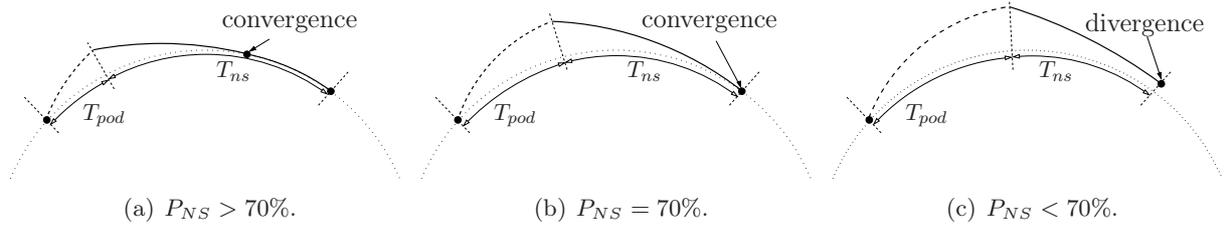


FIGURE 1.24 – Représentation schématique de la convergence de la méthode hybride DNS-POD. \dots cible exacte; $- - -$ POD ROM; $—$ DNS.

1.3.3 Construction d'une base POD robuste : Iterative Centroidal Voronoi Tessellation (ICVT)

Ce travail a été réalisé dans le cadre du projet européen FFAST, et est décrit dans les documents de Bergmann *et al.* (2012a,b).

Configuration d'étude

Le but de cette section est de construire une base POD qui soit robuste vis à vis de paramètres d'entrée du système. La construction de la base POD est effectuée "off-line" et elle n'est donc pas réactualisée. Par exemple, Braconnier *et al.* (2011) ont utilisé une méthode dite *leave-one-out* pour déterminer à quel endroit du sous espace considéré il est nécessaire d'enrichir la base POD. Nous avons opté pour une méthode différente dans cette étude. Le travail de cette dernière section a été effectué lors du projet FP7 FFAST piloté par par l'université de Bristol (UK) avec Airbus UK. Le cas test est un écoulement compressible autour d'un profil d'aile 2D qui oscille sinusoidalement autour de son centre de masse. Les paramètres d'entrée du système sont donc l'amplitude et la fréquence des oscillations. Ce travail à été réalisé avec Haysam Telib (OPTIMAD Engineering <http://www.optimad.it>, spin off du Politecnico di Torino).

Nous avons considéré le même profil d'aile NACA0012 oscillant que dans le rapport AGARD-R702. Le centre de rotation est situé à 25% de la corde. Ce test correspond à un nombre de Mach 0.6 à l'infini. Le nombre de Reynolds dans les expériences est 4.8×10^6 , et nous avons fait l'approximation d'un écoulement sans viscosité. L'espace des paramètres d'entrée est en deux dimensions : la fréquence des oscillations varie entre $f^1 = 30\text{Hz}$ et $f^2 = 70\text{Hz}$ (le cas CT1 : 50Hz) et l'amplitude des oscillations varie entre $\alpha_0^1 = 1.6^\circ$ et $\alpha_0^2 = 3.6^\circ$ (CT1 : 2.5°). Le domaine de calcul est $\Omega = 30c \times 20c$, où c est la corde du profil d'aile. Le profil est positionné avec $10c$ au dessus, $10c$ en dessous, $10c$ en amont et $20c$ en aval. Le maillage utilisé est $(4.8 \times 10^3) \times (4.8 \times 10^3)$. La condition initiale est un écoulement non perturbé uniforme. L'intégration temporelle est poursuivie jusqu'à obtention d'un cycle périodique qui correspond approximativement à deux périodes d'oscillations (voir figure 1.25). Le code de calcul et les simulations numériques sont réalisés avec le code CoCoFlo par Haysam Telib. Des snapshots du nombre de Mach sont représentés sur la figure 1.26.

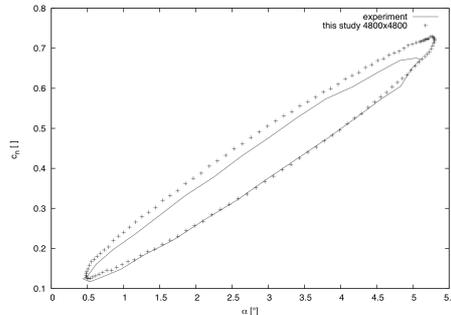


FIGURE 1.25 – Coefficient de force normale en fonction de l'angle. Résultats DNS et expérimentaux d'AGARD R702.

Nous avons alors développé et implémenté un algorithme d'échantillonnage du sous espace engendré par les

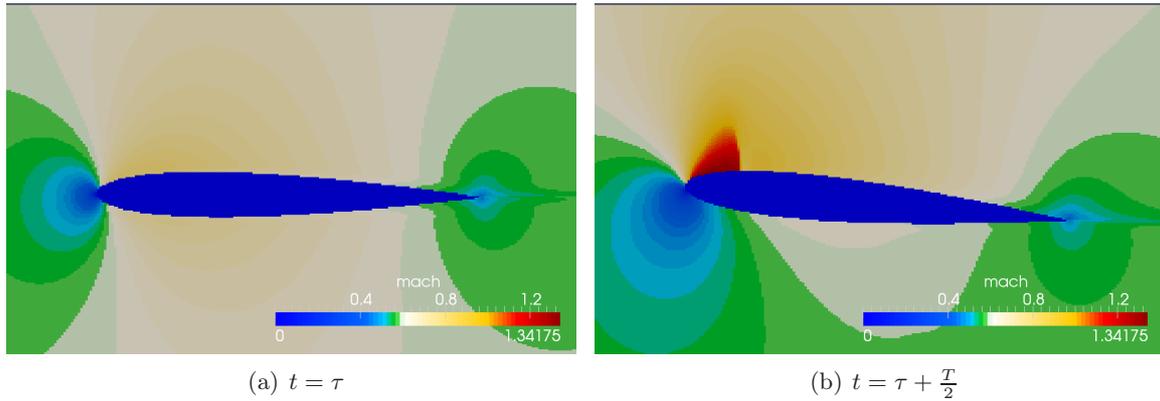


FIGURE 1.26 – Nombre de Mach.

paramètres d'entrée (f et α) afin d'enrichir les fonctions de base POD. L'objectif de cette procédure est de déterminer une base POD qui minimise une erreur d'approximation sur le sous espace $S = [\alpha_0^1, \alpha_0^2] \times [f^1, f^2]$. Pour être plus précis nous construisons 3 bases POD, une pour la vitesse du son a , une autre pour la pression p , et enfin une autre pour la vitesse \mathbf{u} (variable primitive). De même, lorsque nous parlerons d'erreur associée à la base POD, nous étudierions en fait l'erreur moyenne sur les 3 bases⁷. Puisque le profil d'aile bouge en espace (il oscille), nous avons décidé de réaliser toutes les corrélations spatiales sur le domaine entier Ω privé d'un sous domaine rectangulaire entourant le profil d'aile. Un exemple de fonctions de base ainsi obtenues est présenté sur la figure 1.27. Un exemple de spectre des valeurs propres obtenues est représenté sur la figure 1.28

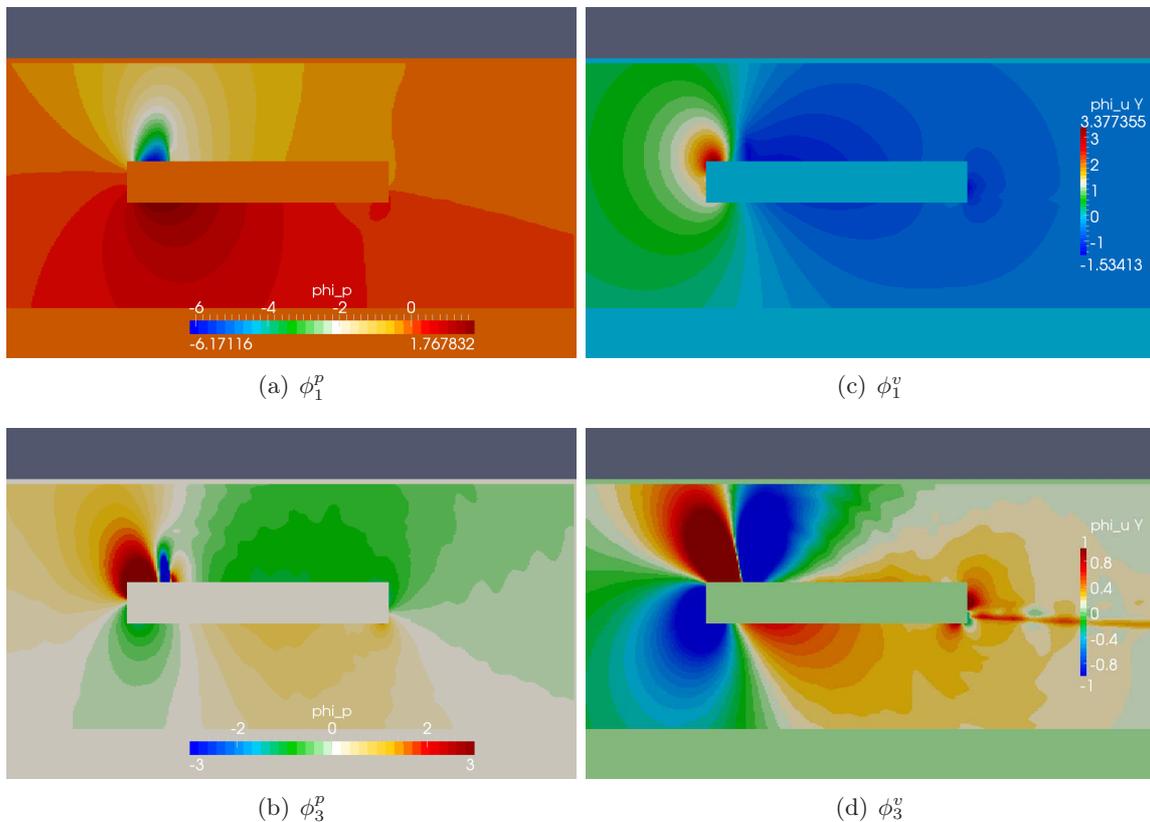


FIGURE 1.27 – Fonctions de base POD pour la pression (gauche) et la vitesse verticale (droite).

⁷. Ceci est tout à fait arbitraire, et des pondérations plus judicieuses pourraient être effectuées au besoin pour donner plus d'importance à certains quantifiés

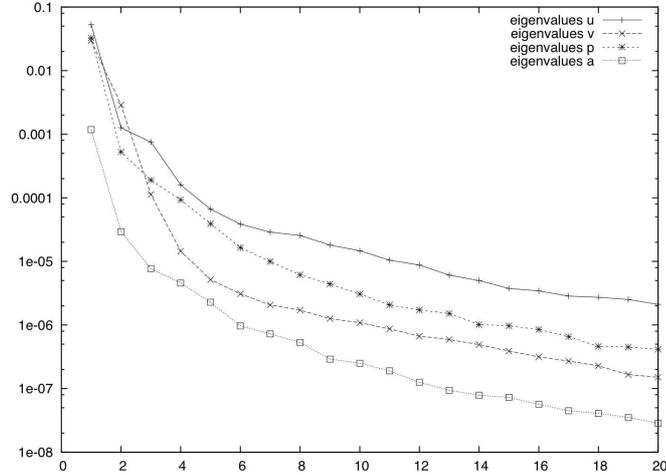


FIGURE 1.28 – Spectre des valeurs propres pour les composantes de vitesse u et v , la pression p et la vitesse du son a .

Méthode

L'idée principale est de construire itérativement un diagramme de Voronoi ainsi que la triangulation de Deulaunay associée⁸ basés sur l'erreur de projection POD. Cette procédure est décrite dans Bergmann *et al.* (2013) et est une extension des travaux que nous avons réalisés dans le cas monodimensionnel (Lombardi *et al.*, 2011). Soit \mathcal{P}_n le jeu de points P_1, \dots, P_n dans l'espace des paramètres qui correspondent aux simulations Navier-Stokes déjà réalisées, et soit \mathcal{T}_n la triangulation de Deulaunay associée. Nous construisons M fonctions de base POD⁹ ϕ_i , $i = 1, \dots, M$ à partir des simulations aux points P_1, \dots, P_n . Nous déterminons ensuite l'erreur de représentation $E(P_k)$, $k = 1, \dots, n$, qui correspond à la norme L^2 des résidus de la projection des solutions détaillées (DNS) aux points P_k sur la base ϕ_i , $i = 1, \dots, M$. Soit $V(T_s)$ la base formée par les points $T_s \in \mathcal{T}_n$. Nous sélectionnons le triangle $T_{\max} \in \mathcal{T}_n$ sur lequel l'erreur $P_k \in V(T_s)$ (le produit de son aire avec la somme des erreurs aux sommets $E(P_k)$) est maximale. Le point suivant de la triangulation est le barycentre de T_{\max} . Avec ce nouveau point, nous construisons une nouvelle triangulation de Deulaunay. A chaque itération nous avons alors besoin de recalculer une nouvelle triangulation de Deulaunay.

Nous réalisons tout d'abord une transformation du sous espace $S = [\alpha_0^1, \alpha_0^2] \times [f^1, f^2]$ sur la carré unité ($\alpha_0 = [\alpha_0^1, \alpha_0^2] \mapsto \overline{A} = [0, 1]$ et $f = [f^1, f^2] \mapsto \overline{F} = [0, 1]$). La figure 1.29 représente une itération donnée de l'algorithme d'enrichissement avec 8 triangles relatifs à 7 points de simulation construits initialement avec les points P_1, P_2, P_3, P_4 . La triangulation de Delaunay (en rouge) et la tessellation de Voronoi (en bleu) sont présentés. Le nouveau point P_8 est ajouté au barycentre du triangle P_2, P_4, P_5 d'erreur maximale. Cette procédure peut être étendue à des sous espaces de paramètres d'entrée de dimensions supérieures¹⁰

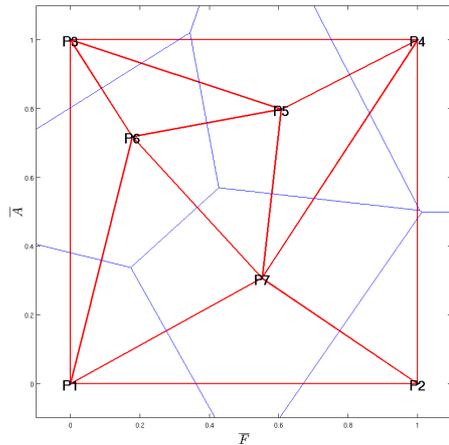
Résultats

Nous commençons le processus d'enrichissement à partir d'une base POD, appelée $B_{Initial}$, construite avec les simulations numériques (DNS) des points P_1, P_2, P_3 et P_4 (voir Fig. 1.30). Nous extrayons 20 réalisations temporelle uniformément réparties sur une période d'oscillation pour chaque point P_i , $1 \leq i \leq 4$.

8. La triangulation de Deulaunay est le dual du diagramme de Voronoi

9. Le nombre de modes POD M est arbitraire, et il est gardé constant dans tout le processus d'enrichissement de la base POD. Une étude de convergence de ce paramètre pourra être nécessaire.

10. Nous savons que la triangulation de Deulaunay existe, même si sa construction est moins triviale à partir d'hypersphères.



(a) itération typique (itération 3)

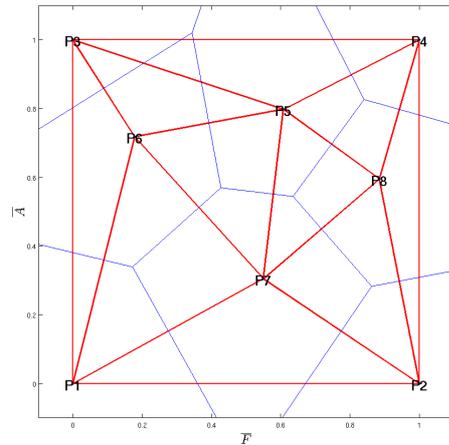
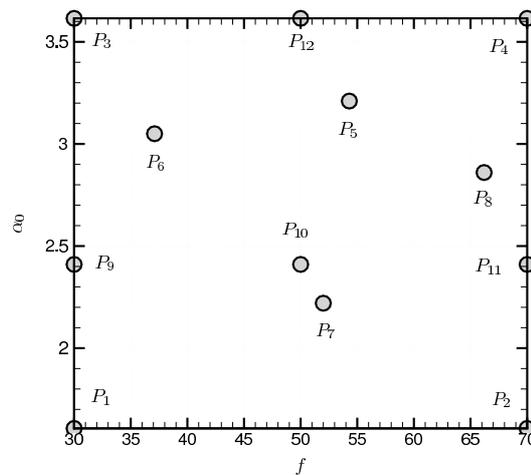
(b) un nouveau point P_8 est ajouté et la triangulation est mise à jourFIGURE 1.29 – Exemple d’une itération du processus d’enrichissement de Voronoi. Le sous espace S est représenté.

FIGURE 1.30 – Echantillonnage des paramètres d’entrée.

A partir de ces points, 4 nouveaux points P_5 , P_6 , P_7 et P_8 ont été rajoutés itérativement avec la méthode décrite précédemment. Une base sous optimale, notée B_{Subopt} , est alors construite avec ces 8 points, P_1 à P_8 . Nous voulons comparer cette base sous optimale avec une autre base construite arbitrairement avec 8 points. Le plus naturel est de construire une base avec des points uniformément répartis sur le sous-espace des paramètres d’entrée. On note $B_{Uniform}$ cette base construite avec les points P_1 à P_4 et les points P_9 à P_{12} où des DNS sont réalisées. La table 1.2 récapitule les points de simulations utilisés pour construire chaque base POD.

La précision des 3 bases POD est évaluée en calculant la norme L^2 de l’erreur de projection des tous les points à disposition (P_1 à P_{12}) sur chaque base POD, voir la table 1.3. L’erreur P_7 représente l’erreur moyenne sur tous les points P_1 à P_{12} . La base B_{Subopt} présente l’erreur la plus faible avec une réduction de l’erreur d’environ 15% par rapport à la base $B_{Uniform}$. Il est également intéressant de noter que les erreurs sur les points P_9 à P_{12} qui ne sont pas inclus dans la base POD B_{Subopt} sont très proches des erreurs obtenues avec $B_{Uniform}$ (qui inclue ces points). La réciproque n’est pas aussi bien vérifiée. Ceci mais en lumière la

base POD	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
$B_{Initial}$ uniforme	X	X	X	X								
$B_{Uniform}$ uniforme	X	X	X	X					X	X	X	X
B_{Subopt} sous optimale	X	X	X	X	X	X	X	X				

TABLE 1.2 – Bases POD utilisées.

robustesse de la base POD obtenue avec la processus d’enrichissement proposé.

E	P_1	P_2	P_3	P_4	P_9	P_{10}	P_{11}	P_{12}	P_5	P_6	P_7	P_8	P_T
$B_{Initial}$	3.71	3.75	7.36	4.80	6.20	5.25	5.58	3.80	4.69	4.53	3.75	4.63	4.82
$B_{Uniform}$	3.85	4.07	6.70	5.29	4.91	4.20	4.87	4.18	4.38	4.29	3.89	4.45	4.60
B_{Subopt}	3.24	3.23	5.42	5.41	5.11	4.62	4.99	4.20	3.74	3.37	3.01	3.06	4.08

TABLE 1.3 – Erreur de projection L^2 ($\times 10^4$). L’erreur moyenne P_T est évaluée sur les 12 points P_i . Les bases $B_{Uniform}$ et B_{Subopt} sont calculées avec 160 snapshots. La base $B_{Initial}$ est calculée avec 80 snapshots.

1.4 Conclusions et perspectives

Le but de ce chapitre était de présenter des méthodes pour améliorer les modèles réduits (ROM) construits par décomposition orthogonale aux valeurs propres (POD). En effet, la POD est une méthode très efficace pour construire des modèles réduits, mais possède également quelques inconvénients : (i) en général, la POD est construite à partir des champs de vitesse et la pression doit être modélisée, (ii) il est nécessaire de modéliser les effets des petites échelles non résolues par le POD ROM dans lesquelles s’effectue une grande partie de la dissipation visqueuse, et (iii) les fonctions de base POD sont uniquement capables de bien représenter la dynamique d’écoulement que présente la base de données (snapshots).

Nous avons proposé une méthode pour construire un modèle réduit POD étendu au champ de pression. La construction de ce modèle n’engendre pas de coûts réellement supérieurs par rapport au modèle classique construit sur les champs de vitesse. L’utilisation de ce modèle à pression permet de s’affranchir de la modélisation du terme de pression dans le ROM. Il est de plus possible d’évaluer les résidus de l’opérateur de Navier-Stokes évalué avec les champs de vitesse et de pression reconstruits par POD. Bien que ce modèle donne de bons résultats, il est tout de même nécessaire de modéliser l’effet des échelles non résolues.

A cet effet, plusieurs méthodes de stabilisation du modèle réduit ont été proposées. L’idée est de modéliser l’effet des échelles non résolues en utilisant les résidus de l’opérateur de Navier-Stokes évalué avec les champs POD. La première méthode consiste à augmenter la taille du sous espace POD en ajoutant quelques modes POD des résidus. Cette méthode permet de stabiliser une base POD qui était nativement instable. Nous avons par exemple montré sur notre cas test qu’un modèle POD à 5 modes n’était pas stable (plus exactement pas précis), mais qu’un modèle à 5 modes avec 3 modes POD classiques plus 2 modes POD de résidus permet d’améliorer significativement la précision du modèle. De plus cette stabilisation n’introduit aucun paramètre empirique. Dans la seconde approche les échelles non résolues sont approchées. Des méthodes de type SUPG et VMS ont été développées et donnent de très bons résultats. Dans ce cas, 2 paramètres doivent être ajustés (ou optimisés). Dans certains cas particuliers, Hugues *et al.* (1998) ont montré que ces paramètres pouvaient être obtenus avec des arguments physiques pour les équations de Navier-Stokes.

Finalement, nous avons proposé des méthodes pour améliorer la représentativité du sous espace propre POD. La première classe de méthodes consiste à adapter itérativement la base POD quand la dynamique de l’écoulement évolue (variation de paramètres d’entrée ou paramètres de contrôle). Nous avons initialement étudié une méthode de Krylov. Cette méthode consiste à ajouter itérativement des modes POD de résidus quand la dynamique évolue (il s’agit de la version itérative de la première méthode de stabilisation du ROM). Quand la taille de la base dépasse un seuil fixé, une nouvelle compression POD est effectuée. Cette méthode donne de très bons résultats pour l’équation de Burgers 1D mais ne rencontre pas le même succès pour les équations de Navier-Stokes 2D où la convergence, si elle existe, est très lente. Une explication est que les

échelles non résolues ne sont pas forcément des petites échelles, et leur approximation sur les résidus n'est peut être pas adaptée. Ensuite, nous avons développé une méthode hybride qui couple POD ROM et DNS. L'idée est de modifier et d'actualiser la base de données quand la dynamique évolue, de telle sorte que la base de donnée soit toujours adaptée à la dynamique. Cette méthode, couplée avec des algorithmes très efficaces pour actualiser les fonctions de base POD, donne de très bons résultats si un certain seuil de DNS est effectué.

La seconde classe de méthode consiste à construire une base globale qui soit robuste vis à vis d'un sous espace de paramètres d'entrée du système de façon à ne pas réactualiser la base au cours d'un éventuel processus d'optimisation. Nous avons déjà développé une méthode pour calculer une base dite généralisée en utilisant une loi de contrôle qui balaye un ensemble de dynamiques dans un sous espace considéré (Bergmann *et al.*, 2005). Cette méthode est très efficace mais nécessite tout de même un certain empirisme dans le choix de la loi de contrôle. Nous avons ici proposé une méthode sans empirisme basée sur la CVT (centroidal voronoi tessellation). Cette base POD est construite itérativement en ajoutant des points dans les zones du sous espace, formé par les paramètres d'entrée, dans lesquelles une approximation de l'erreur est maximale. L'approximation de l'erreur est basée sur les résidus, et la construction de nouveaux points est basée sur une triangulation de Delaunay (dual de la CVT). Cette méthode est très robuste, l'erreur étant approximativement constante sur tout le sous-espace considéré.

Enfin, comme il sera mentionné plus en détail dans la conclusion générale de ce manuscrit, je vais m'intéresser à d'autres types de méthodes de réduction de modèle comme la PGD (Proper Generalized Decomposition). Ceci sera effectué via une thèse que je vais co-encadrer avec Mejdî Azaiez (I2M, Bordeaux) et Angelo Iollo. Cette thèse concerne la simulation numérique du processus décrivant le stockage d'énergie par changement de phase, en partenariat avec la société espagnole ABENGOA¹¹.

Finalement, pour conclure réellement avec les modèles réduits construits par POD, on peut dire plusieurs choses. Tout d'abord, les modèles réduits POD n'ont rien de magique comme on peut le penser au premier abord (du moins comme je l'avais imaginé naïvement en commençant ma thèse). Ils peuvent être très utiles pour la compression de données par exemple, ou pour l'interpolation. Par contre, la prédiction est plus délicate. En effet, nous pouvons rencontrer des problèmes de stabilité et de robustesse comme il a déjà été mentionné. Ensuite, avec un certain recul, nous pouvons dire que les modèles réduits POD peuvent être efficacement utilisés conjointement avec une connaissance *a priori* de la physique étudiée. Il faut donc prendre de nombreuses précautions, que ce soit sur la construction de la base ou sur la construction du modèle. La "magie" de la POD peut opérer sur des problèmes simples (burgers monodimensionnel, par exemple), mais dès que la complexité croît (Navier-Stokes), il n'en va plus de même. Par exemple, la construction de bases POD *a priori* (*i.e.* sans avoir recours à des simulations numériques) est efficace pour de simples problèmes, mais pas pour Navier-Stokes. Après avoir un temps cherché des solutions pour encore améliorer les modèles réduits POD, j'ai décidé de réorienter quelque peu mes activités de recherches, mais sans abandonner les modèles réduits. En effet, la dernière section consacrée à la construction d'une base robuste a été réalisée récemment dans le cadre du projet européen FFAST. Mes activités les plus récentes concernent la modélisation et la simulation numérique en mécanique des fluides, et seront présentées dans les chapitres suivants.

11. <http://www.abengoa.com>

Deuxième partie

Modélisation et simulations numériques
en mécanique des fluides

Introduction et motivations de l'étude

Cette partie a été initiée dès 2009 dans l'ANR CARPEiNTER portée par Angelo Iollo. En particulier, il était question de développer et d'implémenter des méthodes numériques efficaces pour la simulation et le contrôle de phénomènes physiques qui interviennent dans de nombreuses applications. L'application centrale est la mécanique des fluides. Nous nous sommes néanmoins intéressés à des problèmes en aérodynamique, en environnement, et dans un contexte biologique en utilisant un paradigme innovant basé sur l'utilisation de grilles cartésiennes, de pénalisation volumique et frontières immergées, ainsi que de fonctions level set. L'utilisation de grilles cartésiennes permet d'éviter les problèmes de maillage et remaillage pour des géométries complexes, et permet également d'utiliser des schémas numériques simples et précis. La pénalisation volumique, initialement développée à Bordeaux par Angot *et al.* (1999), est une technique efficace pour imposer implicitement les conditions aux limites sur des obstacles immergés. La pénalisation permet d'appréhender le calcul des interactions fluide structures, et des interactions multi physiques en général, pour des coûts de calculs réduits avec une mise en place simplifiée. Les fonctions level set décrivent les géométries d'une façon non paramétrique, et les changements topologiques dus à la nature physique ou à l'optimisation peuvent facilement être pris en compte. Notre objectif est de résoudre des problèmes réalistes en développant un code de calcul qui prend avantage de la structure cartésienne de la grille pour un calcul massivement parallèle.

Le but final est de simplifier la mise en route des simulations numériques. En effet, il est souvent difficile d'initialiser les simulations : on a besoin d'un maillage pour la structure et d'un maillage pour le fluide, qui en général coïncide avec le maillage de la structure. Le temps de développement est alors très important, et à reproduire pour chaque étude où la géométrie change. Notre objectif est de faire un premier pas vers ce que l'on nomme *Drag And Drop Simulation*. On veut remplacer le temps de développement ingénieur pour la mise en route des simulations par du temps machine en profitant des nouvelles architectures massivement parallèles des clusters de calcul.

Cette partie se compose de trois chapitres (chapitres 2, 3 et 4), dont le dernier reste encore dans un stade exploratoire.

Le chapitre 2 a pour but de présenter le modèle physique étudié, à savoir les équations de Navier-Stokes incompressibles, et les méthodes numériques communes utilisées dans les chapitres suivants. Au besoin, des méthodes numériques spécifiques sont introduites dans chaque chapitre. En particulier nous présenterons le suivi et la capture d'interface §2.2 avec notamment les méthodes utilisant les fonctions level set, et plus particulièrement la fonction distance signée §2.3. Nous présenterons ensuite les méthodes de modélisation des écoulements fluides en présence d'interfaces fluide/structure, fluide/fluide voire fluide/fluide/structure §2.4. Une fois que nous aurons établi les différents modèles, nous proposerons des méthodes numériques pour résoudre les interfaces fluide/structure §2.5, les interfaces fluide/fluide §2.6, et les interfaces fluide/fluide/structure §2.7. Enfin, le code de calcul NaSCar¹² est présenté et brièvement validé §2.8. Les méthodes spécifiquement développées pour les interactions fluide/structure sont publiées dans Bergmann et Iollo (2011) et Bergmann *et al.* (2014b)

Le chapitre 3 présente une application d'écoulement autour d'obstacles qui se déforment et se déplacent dans le fluide : le déplacement autopropulsé de nageurs de type poissons. Dans tout ce chapitre, nous imposons une nage paramétrique §3.2 en déformant la ligne médiane du corps que nous appellerons colonne vertébrale. Dans tout ce qui suit, nous nous intéresserons aux mécanismes de la nage, et plus particulièrement à des aspects liés à des considérations énergétiques de la nage. Nous avons naturellement commencé par étudier des nageurs plongés dans un écoulement bi-dimensionnel §3.3. Nous avons en particulier comparé l'efficacité énergétique pour différents nageurs (différents profils et différentes lois de nage). Nous nous sommes également intéressés à la nage en groupe ou la nage intermittente. Après cette entrée en matière, nous nous sommes intéressés à des nageurs de type poissons dans un écoulement tridimensionnel §3.3. L'efficacité de la nage en fonction de la loi imposée a été considérée. Nous nous sommes ensuite concentré sur la rigidité ou flexibilité de la nageoire caudale. Nous avons observé que, pour certains paramètres de rigidité, l'efficacité peut être améliorée de plus de 30%. Différents prototypes de nageurs tels que des méduses ou des raies ont également été brièvement étudiés. Enfin, nous nous sommes intéressés à ce que l'on appelle *Drag And Drop Simulation*. Ceci consiste à effectuer des simulations numériques à partir d'un corps défini par des images (photographies). A partir de photographies d'un poisson, nous calculons le squelette (avec une technique de

12. Navier-Stokes Cartésien

squelettisation en collaboration avec Cédric Galusinski et Christian N’Guyen de l’IMATH à Toulon) puis nous reconstruisons des profils associés à une loi de nage. Le champ de déplacement d’une image à l’autre est déterminé par transport optimal. Enfin, à partir de ces profils et champs de déplacement, nous pouvons effectuer une simulation numérique. Une partie des résultats de ce chapitre est publiée dans Bergmann et Iollo (2011), Bergmann *et al.* (2014b) et dans le proceeding Bergmann *et al.* (2014a).

Le chapitre 4 présente les premiers résultats obtenus pour des systèmes de récupération de l’énergie. Ce chapitre est encore dans un état relativement exploratoire. La première application est toujours un obstacle qui se déplace dans un fluide : il s’agit d’une éolienne. Cette étude a été initiée en 2010 par une collaboration avec la PME VALEOL¹³, filière du groupe VALOREM¹⁴. VALEOL est une société localisée à Bègles en Gironde. Un de leurs objectifs est d’optimiser des pales d’éoliennes, et en particulier d’étudier le potentiel bénéfique d’ajouter des appendices aux pales existantes. Je co encadre actuellement la thèse CIFRE de Xin Jin. Le but de ce projet est de réaliser une chaîne de conception pour optimiser la forme d’une pale. Les coûts de calculs associés à une simulation tridimensionnelle d’une turbine éolienne sont très élevés et ne permettent pas d’effectuer une optimisation via une simulation haute fidélité. Nous avons alors opté pour une modèle simplifié bidimensionnel axisymétrique. Le code associé s’appelle VALDAG¹⁵ et a été développé dans le cadre de la thèse de Xin. Ce code, avec une interface graphique java, permet de calculer (via quelques approximations) la puissance que peut extraire le rotor pour différentes vitesses de vent. Il est également possible d’optimiser la pale pour répondre aux attentes industrielles. Actuellement l’optimisation agit uniquement sur le l’angle d’attaque pour toute section de la pale. Idéalement, la pale à optimiser sera testée par le solveur 3D NaSCar. Beaucoup de travail reste cependant à accomplir sur ce solveur pour le rendre fiable pour des simulations avec des nombres de Reynolds très élevés, et donc des couches limites turbulentes qu’il est actuellement impossible de calculer avec la méthode décrite au chapitre 2. La seconde application est encore plus récente et consiste à extraire l’énergie de la houle par des structures articulées en forme de serpents de mers. Cette étude est réalisée en collaboration avec Lisl Weynans (IMB) et Francky Luddens (post-doc INRIA, IMB) sur l’interaction bi fluide et level set, ainsi qu’avec Pierre Lubin (I2M, Bordeaux) pour son expertise de la houle. Cette application couple deux fluides (air et eau) avec une structure articulée qui est élastique. Avant d’arriver à cette application nous avons testé le solveur qui couple un fluide avec une structure, puis le solveur bi fluide. Des validations restent cependant à effectuer. Certains résultats de ce chapitre sont publiés dans les proceedings Bergmann et Iollo (2012), Jin *et al.* (2012) et Bergmann *et al.* (2014c).

13. <http://www.valeol.fr>

14. <http://www.valorem-energie.com>

15. http://www.valeol.fr/valdag_web

Chapitre 2

Modélisation et méthodes numériques pour des fluides en présence d'interfaces

Sommaire

2.1	Introduction	41
2.2	Suivi et capture d'interface : choix du maillage	42
2.2.1	Approches Lagrangiennes : suivi d'interfaces	42
2.2.2	Approches Eulériennes : capture d'interface	43
2.3	Fonction level set : la fonction distance signée	44
2.3.1	Transport de la fonction level set	45
2.3.2	Redistanciation	47
2.3.3	Conservation de la masse	51
2.4	Modélisation d'écoulements fluides en présence d'interfaces	52
2.4.1	Modèle avec interfaces fluide/fluide/structure	52
2.4.2	Modèle avec interface fluide/structure	54
2.4.3	Modèle avec interface fluide/fluide	54
2.5	Domaine fictif et interface fluide/structure	55
2.5.1	Configuration d'écoulement et modèle de Navier-Stokes incompressible	55
2.5.2	Approche continue	55
2.5.3	Approche discrète	56
2.5.4	Une méthode hybride au second ordre	58
2.5.5	Résolutions numériques des équations	58
2.6	Modélisation et résolution numérique des interfaces fluide/fluide	62
2.6.1	Interface lisse et <i>Continuum Surface Force</i> (CSF)	62
2.6.2	Interface raide et <i>Ghost Fluid Method</i> (GFM)	63
2.7	Interfaces fluide/fluide/structure et ligne triple	64
2.8	Le code de calcul NaSCar (NAvier-Stokes CARtésien)	64
2.8.1	Description	64
2.8.2	Validations	65
2.9	Conclusions	66

2.1 Introduction

Ce chapitre a pour but de présenter les modèles étudiés ainsi que les méthodes numériques associées pour résoudre différents problèmes issus de la mécanique des fluides. Le point central est de modéliser et de simuler des problèmes faisant intervenir des interfaces entre différents milieux : on considèrera des interfaces fluide/structure et fluide/fluide ou encore fluide/fluide/structure. De grandes déformations peuvent

être présentes dans le cas d’une interface entre deux fluides, et il est donc nécessaire de pouvoir suivre ce type d’interfaces de façon précise. Le paragraphe §2.2 présentera deux classes de méthodes qui permettent de suivre précisément les interfaces. La première est basée sur un suivi lagrangien où un remaillage est nécessaire à chaque instant. La seconde est basée sur une méthode eulérienne qui permet de représenter l’interface sur un maillage fixe, comme un maillage cartésien par exemple. En particulier, nous utiliserons des fonctions level set tout au long de ce manuscrit §2.3. Nous présenterons ensuite quelques modèles d’écoulements fluides pour différents types d’interfaces §2.4 avec les conditions aux limites associées. Une fois ces modèles présentés, nous proposerons des méthodes numériques pour résoudre ce type d’écoulements sur des maillages cartésiens avec des interfaces fluide/structure §2.5, des interfaces fluide/fluide §2.6, puis des interfaces fluide/fluide/structure §2.7. Enfin, le code de calcul NaSCar sera brièvement présenté §2.8

2.2 Suivi et capture d’interface : choix du maillage

Deux approches peuvent être utilisées pour représenter une interface entre deux phases différentes (que ce soit fluide/solide ou fluide/fluide). La première consiste en un suivi Lagrangien de marqueurs (particules) sans masses initialement répartis sur l’interface. Cette technique est adaptée pour représenter une interface évolutive fluide/solide pour des petites déformations. Nous utiliserons cette technique dans la suite sur certains exemples, notamment dans les chapitres 3 et 4. Une autre approche est de représenter implicitement l’interface à l’aide d’une fonction scalaire. La capture de l’interface est identifiée à l’iso ligne zéro de cette fonction scalaire. Cette méthode Eulérienne permet de modéliser les grandes déformations qui peuvent intervenir lorsque l’on souhaite suivre une interface fluide/fluide comme présenté au chapitre 4. Cette fonction peut prendre différentes formes comme un champ de phase ou une fonction level set. Dans la suite nous utiliserons principalement une fonction level set qui est la distance signée à l’interface. Ce choix repose principalement sur les propriétés mathématiques que présente la fonction distance.

2.2.1 Approches Lagrangiennes : suivi d’interfaces

L’interface fluide/fluide ou fluide/structure peut être suivie par des marqueurs ou particules sans masses. Une particule de position \mathbf{x}_i est transportée par le champ de vitesse local \mathbf{u}_i par

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_i. \quad (2.1)$$

Suivant le type de méthode utilisé, la vitesse \mathbf{u}_i sera soit interpolée sur un maillage fluide qui ne coïncide pas avec les particules, soit directement évaluée sur le nœud du maillage fluide si celui ci coïncide avec les particules.

Les marqueurs de volume basés sur la méthode Marker And Cells (MAC)

La méthode MAC a été initialement développée par Harlow et Welch (1965) pour simuler des écoulements visqueux incompressibles à surface libre, *i.e.* en présence d’une seule phase. Le fluide est résolu dans un domaine Ω discrétisé sur une grille fixe, en général cartésienne. Le fluide occupe uniquement un sous domaine Ω_f délimité par l’interface Γ (voir figure 2.1(a)). Des marqueurs sont initialement (uniformément) répartis dans Ω_f et transportés via l’équation (2.1). L’interface est alors déterminée sur le maillage fluide par la présence ou non de marqueurs d’une cellule à l’autre. Cette méthode a ensuite été beaucoup étudiée, et des développements vers des problèmes diphasiques ont été apportés peu après par Daly (1967). Cette méthode est performante pour déterminer les séparations ou jonctions de l’interface, ainsi que pour conserver le volume. Cependant, puisque l’interface est identifiée par la présence ou non de marqueurs d’une cellule fluide à l’autre, il est nécessaire d’introduire beaucoup de marqueurs en comparaison du nombre de cellules fluides. De plus, en présence de grandes déformations, il est également nécessaire de redistribuer (de remailler) ces marqueurs sur la grille fluide. Le temps de calcul associé à cette méthode peut alors devenir rapidement prohibitif, notamment pour des problèmes en trois dimensions.

Les marqueurs de front

La méthode des marqueurs de front est une extension des marqueurs de volume bifluide qui a été initialement proposée par Daly (1968). Cette méthode propose de distribuer et suivre des marqueurs sur l’interface uniquement (voir figure 2.1(b)). De cette manière, l’interface peut être calculée très précisément en reliant

les marqueurs voisins par des splines de degrés élevés. Bien qu'étant *a priori* attractive, cette méthode possède également quelques défauts. Il n'est pas aisé de simuler des coalitions et ruptures de l'interface. Il peut également être nécessaire de redistribuer les marqueurs sur l'interface. De plus, l'extension aux écoulements 3D n'est pas immédiate.

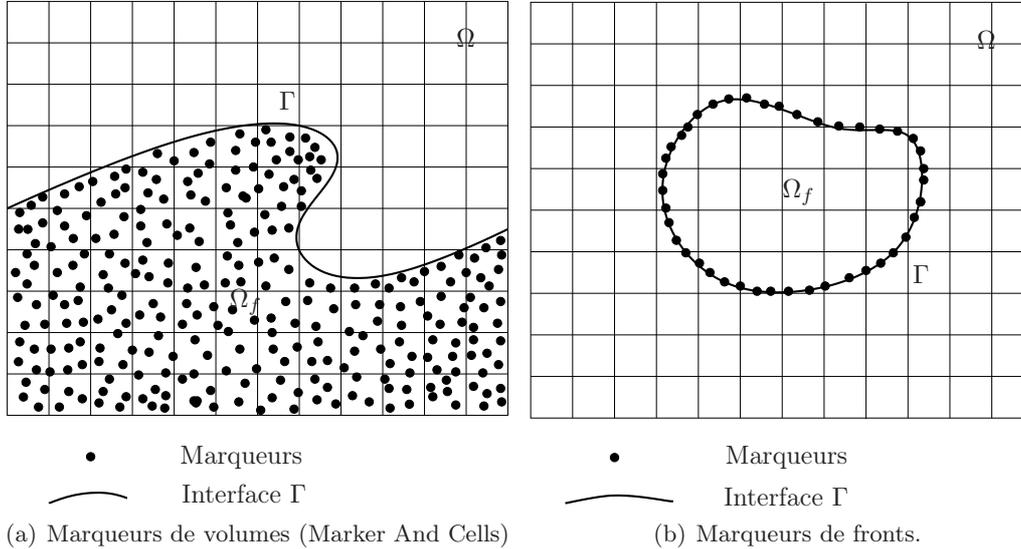


FIGURE 2.1 – Représentation schématique de la méthode de marqueurs.

2.2.2 Approches Eulériennes : capture d'interface

L'idée des méthodes eulériennes de capture d'interface est d'utiliser un champ scalaire, ayant des valeurs différentes dans chaque phase, qui permet de reconstruire implicitement l'interface. Ce champ scalaire, notée ϕ par exemple, est alors transporté par la vitesse locale $\mathbf{u}(\mathbf{x}, t)$ de l'écoulement¹ :

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \quad (2.2)$$

Si le fluide est incompressible ($\nabla \cdot \mathbf{u} = 0$), l'équation (2.2) peut se réécrire sous la forme :

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) = 0. \quad (2.3)$$

Ce champ est capable de gérer aisément les ruptures d'interface, et dans certains cas de conserver des propriétés utiles telles que la masse, ou de permettre un calcul précis de la courbure. Plusieurs méthodes sont couramment utilisées comme la méthode Volume Of Fluid (VOF), la méthode level set ou la méthode des champs de phase. Nous allons présenter succinctement les deux premières, puis nous développerons dans la section suivant la méthode level set que nous avons régulièrement utilisée dans cette étude.

La méthode Volume Of Fluid (VOF)

La méthode VOF utilise une fonction couleur ϕ (fonction masque), qui vaut 1 dans la phase définie par le domaine Ω_f (voit figure 2.1(b)) et 0 dans l'autre phase. Au cours de l'intégration numérique de l'équation (2.2) (ou (2.3)), la valeur du champ ϕ pourra être $0 < \phi < 1$ sur les cellules traversées par l'interface. Le volume d'une phase peut être évalué de façon précise (si un schéma numérique adapté est utilisé pour discrétiser l'équation (2.3)). En effet, l'équation (2.3) est une équation de conservation de la quantité ϕ qui représente ici la proportion du volume sur chaque cellule. Pour la phase initialement à $\phi = 1$, le volume est simplement obtenu en intégrant ϕ sur tout le domaine de calcul. L'inconvénient est qu'il est difficile de localiser de façon précise la position de l'interface sur une cellule du maillage avec la caractéristique $0 < \phi < 1$. On sait

1. En général, il suffit que le champ \mathbf{u} coïncide avec le champ de vitesse du fluide sur l'interface.

uniquement qu'il y a un peu de chaque phase. Essentiellement, deux méthodes de reconstruction de l'interface sont utilisées : les méthodes SLIC et PLIC. La méthode SLIC étant moins précise que la méthode PLIC, elle est moins utilisée. Par contre, l'application de la méthode PLIC pour des géométries en trois dimensions peut être complexe. Dans tous les cas, l'interface n'est pas reconstruite de façon très précise, et le calcul du gradient normal ou de la courbure n'est pas directement envisageable. Les méthodes level set ont alors été introduites en partie pour palier ce problème.

La fonction Level Set

Cette méthode a pour but de résoudre les problèmes de la méthode VOF, en particulier de résoudre précisément l'interface. L'idée est de transporter un champ scalaire ainsi que certaines propriétés liées à ce champ. Par exemple, il peut être intéressant de transporter des propriétés de l'interface telles que la normale ou la courbure. En particulier, la fonction distance signée permet de reconstruire ces propriétés de façon précise en prenant quelques précautions de modélisation et de résolution numérique. Les caractéristiques de cette fonction vont être détaillées dans la section suivante.

2.3 Fonction level set : la fonction distance signée

Cette méthode a été originellement introduite par Osher et Sethian (1988) pour apporter des solutions pour une reconstruction précise de l'interface. Nous avons vu à la section précédente que la méthode VOF, largement utilisée pour modéliser et calculer des écoulements diphasiques, ne permet pas une reconstruction précise de l'interface. Osher et Sethian (1988) ont alors introduit la méthode PSC pour "Propagating Surface under Courbature". Ils ont remarqué qu'il était intéressant d'un point de vue géométrique d'utiliser des fonctions à ligne de niveau (*i.e.* $\phi(\mathbf{x}) = c$) pour calculer la courbure. Soit un domaine Ω qui est divisé en deux sous domaines Ω^+ et Ω^- par une interface Γ : une fonction level set est par exemple définie par :

$$\begin{cases} \phi(\mathbf{x}) = 0 \text{ si } \mathbf{x} \in \Gamma, \\ \phi(\mathbf{x}) > 0 \text{ si } \mathbf{x} \in \Omega^+, \\ \phi(\mathbf{x}) < 0 \text{ si } \mathbf{x} \in \Omega^-. \end{cases} \quad (2.4)$$

La fonction ϕ peut être choisie en fonction des besoins de l'utilisateur et des propriétés désirées. De part sa régularité mathématique, la fonction distance signée a alors été introduite. Le signe est arbitraire et est choisi par l'utilisateur. Soit $d(\mathbf{x}) \geq 0$ la distance minimum d'un point quelconque \mathbf{x} à l'interface Γ . L'interface Γ est alors l'iso ligne zéro de la fonction distance signée, *i.e.* $\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\}$ et est définie par :

$$\begin{cases} \phi(\mathbf{x}) = 0 \text{ si } \mathbf{x} \in \Gamma, \\ \phi(\mathbf{x}) = +d \text{ si } \mathbf{x} \in \Omega^+, \\ \phi(\mathbf{x}) = -d \text{ si } \mathbf{x} \in \Omega^-. \end{cases} \quad (2.5)$$

Propriétés mathématiques de la fonction level set

En plus de permettre de capturer facilement l'interface $\Gamma = \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\}$, les fonctions level set possèdent également quelques propriétés mathématiques intéressantes.

- La normale \mathbf{n} à l'interface ainsi que la courbure κ peuvent être évaluées directement par :

$$\mathbf{n} = \frac{\nabla \phi}{\|\nabla \phi\|}, \quad (2.6)$$

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right). \quad (2.7)$$

- De plus, la fonction distance signée est hyper régulière :

$$\|\nabla \phi(\mathbf{x})\| = 1, \quad \forall \mathbf{x}. \quad (2.8)$$

En plus de ces propriétés mathématiques, les fonctions level set sont facilement transposables à des problèmes de dimensions supérieures (le passage du 2D au 3D est immédiat), et l'expérience montre que les connexions et ruptures sont bien prises en compte.

2.3.1 Transport de la fonction level set

Les fonctions level set peuvent être transportées à l'aide de l'équation de transport (2.2). La forme conservative (2.3) pourra également être considérée lorsque $\nabla \cdot \mathbf{u} = 0$. Les schémas numériques pour résoudre cette équation de transport doivent être extrêmement précis en temps et en espace. Des schémas classiques de type Euler, Adams-Basforth en temps, ou centrés voir décentré upwind d'ordre 3 en espace ne sont en général pas suffisants. Les schémas doivent être (i) robustes, (ii) peu diffusifs et (iii) très peu dispersifs. Pour répondre à cette demande, on utilise des schémas Essentiellement Non Oscillants (ENO) et Weighted ENO (WENO) pour la discrétisation en espace ainsi que des schémas TVD (total variation diminishing) Runge-Kutta pour la discrétisation en temps.

Nous considérons l'équation (2.2) en trois dimensions (x, y, z) avec $\mathbf{u} = (u, v, w)^t$, *i.e.* :

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} + w \frac{\partial \phi}{\partial z} = 0. \quad (2.9)$$

Nous considérons de plus une discrétisation spatiale sur un maillage cartésien de pas $\Delta \mathbf{x} = (\Delta x, \Delta y, \Delta z)^t$, l'avancement en temps se fera par pas Δt . Notons de façon général $\phi_{i,j,k}^n = \phi(x_i, y_j, z_k, t^n)$ la fonction discrète en espace et en temps.

Discrétisation en espace : upwind et WENO 5

Le schéma le plus intuitif pour résoudre une équation de transport (ou d'advection) et est schéma décentré de type upwind. Ce schéma consiste à calculer le gradient $\nabla \phi_{i,j,k}$ en choisissant un stencil de discrétisation d'un coté ou de l'autre en fonction du sens de la vitesse $\mathbf{u}_{i,j,k}$. Par exemple,

– si $u_{i,j,k} > 0$, nous choisissons

$$\frac{\partial \phi_{i,j,k}}{\partial x} \approx D_x^- \phi_{i,j,k} = \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x},$$

– si $u_{i,j,k} < 0$, nous choisissons

$$\frac{\partial \phi_{i,j,k}}{\partial x} \approx D_x^+ \phi_{i,j,k} = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}.$$

Les autres directions spatiales sont traitées de façon identique.

Ce schéma est numériquement stable, mais également très dissipatif. Ceci peut avoir pour conséquence une modification de la position de l'interface (déplacement artificiel de l'iso ligne $\phi = 0$), ce que l'on souhaite absolument éviter. L'utilisation d'un schéma d'ordre plus élevé et moins dissipatif est alors indispensable.

Nous introduisons alors un schéma d'ordre élevé qui est le schéma Weight Essentially Non-Oscillatory (WENO) d'ordre 5. Les schémas WENO sont basés sur des schémas ENO (Essentially Non-Oscillatory) introduits par Harten *et al.* (1987). Le principe de ces schémas est de proposer plusieurs stencils de discrétisation (3 en général) pour finalement choisir le stencil le plus régulier afin d'éviter les oscillations qui peuvent se produire notamment dans les zones où la solution varie fortement. Malheureusement, le choix du stencil le plus lisse peut dans certains cas dégrader la précision de la solution près des chocs comme l'ont montré Liu *et al.* (1994). Au lieu de retenir uniquement le stencil le plus régulier Liu *et al.* (1994) proposent un schéma d'ordre 4 qui utilise une combinaison convexe de tous les stencils à disposition, chaque stencil ayant un poids qui définit sa contribution (les poids sont compris entre 0 et 1 et leur somme vaut 1). Ce schéma a alors été amélioré par Jiang et Shu (1996) pour atteindre l'ordre 5. Des ordres supérieurs ont par la suite été développés, mais l'ordre 5 est largement suffisant pour nos applications.

Comme pour le schéma upwind, nous allons raisonner pour la direction x , l'extension aux autres directions est immédiate.

Concrètement, le schéma ENO 3 choisit entre les 3 discrétisations suivantes :

$$\begin{aligned} D_x^\pm \phi_{i,j,k} \Big|_1 &= +\frac{1}{3}v_1^\pm - \frac{7}{6}v_2^\pm + \frac{11}{6}v_3^\pm, \\ D_x^\pm \phi_{i,j,k} \Big|_2 &= -\frac{1}{6}v_2^\pm + \frac{5}{6}v_3^\pm + \frac{1}{3}v_4^\pm, \\ D_x^\pm \phi_{i,j,k} \Big|_3 &= +\frac{1}{3}v_3^\pm + \frac{5}{6}v_4^\pm - \frac{1}{6}v_5^\pm, \end{aligned} \quad (2.10)$$

avec :

$$\begin{aligned}
v_1^- &= \frac{\phi_{i-2,j,k} - \phi_{i-3,j,k}}{\Delta x}, & v_1^+ &= \frac{\phi_{i+3,j,k} - \phi_{i+2,j,k}}{\Delta x}, \\
v_2^- &= \frac{\phi_{i-1,j,k} - \phi_{i-2,j,k}}{\Delta x}, & v_2^+ &= \frac{\phi_{i+2,j,k} - \phi_{i+1,j,k}}{\Delta x}, \\
v_3^- &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x}, & v_3^+ &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}, \\
v_4^- &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}, & v_4^+ &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x}, \\
v_5^- &= \frac{\phi_{i+2,j,k} - \phi_{i+1,j,k}}{\Delta x}, & v_5^+ &= \frac{\phi_{i-1,j,k} - \phi_{i-2,j,k}}{\Delta x}.
\end{aligned} \tag{2.11}$$

Le schéma WENO 5 quant à lui utilise une combinaison convexe des trois dérivées ci-dessus, *i.e.*

$$D_x^\pm \phi_{i,j,k} = w_1^\pm D_x^\pm \phi_{i,j,k}|_1 + w_2^\pm D_x^\pm \phi_{i,j,k}|_2 + w_3^\pm D_x^\pm \phi_{i,j,k}|_3, \tag{2.12}$$

avec la contrainte $w_1 + w_2 + w_3 = 1$. Usuellement, ces poids sont calculés de la manière suivante :

$$w_1^\pm = \frac{a_1^\pm}{a_1^\pm + a_2^\pm + a_3^\pm}, \quad w_2^\pm = \frac{a_2^\pm}{a_1^\pm + a_2^\pm + a_3^\pm}, \quad w_3^\pm = \frac{a_3^\pm}{a_1^\pm + a_2^\pm + a_3^\pm}, \tag{2.13}$$

avec

$$a_1^\pm = \frac{1}{10} \frac{1}{(\epsilon + S_1^\pm)^2}, \quad a_2^\pm = \frac{6}{10} \frac{1}{(\epsilon + S_2^\pm)^2}, \quad a_3^\pm = \frac{3}{10} \frac{1}{(\epsilon + S_3^\pm)^2}. \tag{2.14}$$

Le paramètre ϵ est arbitraire est généralement pris très petit, $\epsilon < 10^{-6}$, et permet d'éviter que le dénominateur soit nul. Les coefficients S^\pm sont donnés par :

$$\begin{aligned}
S_1^\pm &= \frac{12}{13}(v_1^\pm - 2v_2^\pm + v_3^\pm) + \frac{1}{4}(v_1^\pm - 4v_2^\pm + 3v_3^\pm), \\
S_2^\pm &= \frac{12}{13}(v_2^\pm - 2v_3^\pm + v_4^\pm) + \frac{1}{4}(v_2^\pm - v_4^\pm), \\
S_3^\pm &= \frac{12}{13}(v_3^\pm - 2v_4^\pm + v_5^\pm) + \frac{1}{4}(3v_3^\pm - 4v_4^\pm + v_5^\pm).
\end{aligned} \tag{2.15}$$

Cette procédure est répétée pour les autres directions spatiales.

Discrétisation en temps

Pour certaines applications, notamment les écoulements qui présentent de très basses vitesses en micro fluide notamment, un schéma d'intégration temporel à l'ordre un de type Euler explicite peut s'avérer suffisant (Tancogne, 2007). L'équation de transport (2.2) peut se réécrire sous la forme condensée suivante :

$$\frac{\partial \phi}{\partial t} = \mathcal{L}(\phi). \tag{2.16}$$

Le schéma d'Euler est :

$$\phi^{n+1} = \phi^n + \Delta t \mathcal{L}(\phi^n). \tag{2.17}$$

Ce schéma est stable sous la condition CFL (Courant-Friedrichs-Levy) :

$$\Delta t < \overline{\Delta t} = \frac{1}{\frac{\max(|u|)}{\Delta x} + \frac{\max(|v|)}{\Delta y} + \frac{\max(|w|)}{\Delta z}}. \tag{2.18}$$

Cependant, pour les applications considérées dans ce manuscrit, il nous a paru nécessaire d'utiliser un schéma d'ordre supérieur et TVD (total variation diminishing). Shu et Osher (1988) ont donné la forme générale des schémas de Runge-Kutta qui est la suivante :

$$\begin{aligned}
\phi^{(i)} &= \sum_{k=1}^{i-1} \alpha_{ik} \phi^{(k)} + \Delta t \beta_{ik} \mathcal{L}(\phi^{(k)}), \quad i = 1, \dots, m \\
\phi^{(0)} &= \phi^{(n)}, \quad \phi^{(m)} = \phi^{(n+1)}.
\end{aligned} \tag{2.19}$$

Pour $\alpha_{ik} \geq 0$ et $\beta_{ik} \geq 0 \forall i, k$ le schéma de Runge-Kutta est un combinaison convexe de schémas d'Euler . La condition de stabilité est (Shu et Osher, 1988) :

$$\Delta t < c\overline{\Delta t}, \quad \text{avec} \quad c = \min_{i,k} \frac{\alpha_{ik}}{\beta_{ik}} \quad (2.20)$$

Gottlieb et Shu (1998) ont montré, sous les conditions $\alpha_{ik} \geq 0$ et $\beta_{ik} \geq 0 \forall i, k$, que le schéma optimal Runge-Kutta TVD est le suivant :

$$\begin{aligned} \tilde{\phi}^{(n+1)} &= \phi^{(n)} + \Delta t \mathcal{L}(\phi^{(n)}), \\ \tilde{\phi}^{(n+\frac{1}{2})} &= \frac{3}{4}\phi^{(n)} + \frac{1}{4}\tilde{\phi}^{(n+2)} \quad \text{avec} \quad \tilde{\phi}^{(n+2)} = \tilde{\phi}^{(n+1)} + \Delta t \mathcal{L}(\tilde{\phi}^{(n+1)}), \\ \phi^{(n+1)} &= \frac{1}{3}\phi^{(n)} + \frac{2}{3}\tilde{\phi}^{(n+\frac{3}{2})} \quad \text{avec} \quad \tilde{\phi}^{(n+\frac{3}{2})} = \tilde{\phi}^{(n+\frac{1}{2})} + \Delta t \mathcal{L}(\tilde{\phi}^{(n+\frac{1}{2})}). \end{aligned} \quad (2.21)$$

La condition CFL est alors $c = 1$ dans l'équation (2.20).

Le couplage d'une méthode WENO 5 TVD avec une méthode de RK3 TVD permet de s'affranchir de toutes oscillations parasites. Une méthode de RK3 conservative peut également être implémentée, mais nous nous sommes contentés de la méthode RK3 classique, non conservative.

2.3.2 Redistanciation

Généralités

La propriété de fonction distance n'est *a priori* pas conservée lors du transport, sauf pour quelques cas particuliers dans lesquels la vitesse \mathbf{u} le permet. Le principe de la redistanciation est, à partir d'une fonction ϕ^0 qui définit implicitement une interface $\Gamma = \{\mathbf{x} \in \Omega \mid \phi^0(\mathbf{x}) = 0\}$ avec $|\nabla \phi^0| \neq 1$, de déterminer la fonction ϕ telle que $\Gamma = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) = 0\}$ avec $|\nabla \phi| = 1$. La propriété de distance

$$|\nabla \phi| = 1 \quad (2.22)$$

est utile à plusieurs égards. Le fait d'avoir une fonction distance permet de déterminer facilement la courbure de l'interface par exemple. De plus, la fonction distance étant hyper régulière, elle permet d'éviter les erreurs numériques qui pourraient intervenir par transport de forts gradients. Afin d'être certain d'obtenir une fonction distance après chaque étape de transport (ou de temps à autre lors de la simulation), il est nécessaire d'appliquer un algorithme de redistanciation. Ces algorithmes consiste, à partir de la ligne de niveau zéro, de modifier les autres lignes de niveau pour obtenir la condition (2.22).

Le premier algorithme, introduit par Sethian en 1996, et amélioré en 1999, est le fast-marching. Cet algorithme est très rapide comme l'indique son nom, mais est uniquement à l'ordre 1. La ligne de niveau zéro peut donc être modifiée de la taille d'une maille, ce qui est trop important pour nos applications. Des extensions à l'ordre deux et trois ont été introduites par Ahmed *et al.* (2011).

Cependant, nous avons opté pour une méthode itérative basée sur l'idée originale de Sussman *et al.* (1994). Le principe est de résoudre une équation aux dérivées partielles afin de modifier les lignes de niveaux non nulles pour obtenir la condition (2.22). L'équation instationnaire à résoudre est :

$$\begin{aligned} \frac{\partial \phi}{\partial \tau} + \text{sign}(\phi^0)(|\nabla \phi| - 1) &= 0, \\ \phi(\mathbf{x}, \tau = 0) &= \phi^0 = \phi(\mathbf{x}, t). \end{aligned} \quad (2.23)$$

Cette équation est résolue à un temps de la simulation t pour $0 \leq \tau \leq \tau_F$. On souhaite obtenir une solution $\phi(\mathbf{x}, \tau)$ qui converge vers $|\nabla \phi| = 1$, au moins proche de l'interface, pour $\tau = \tau_F$. La fonction signe associée est ² :

$$\text{sign}(\phi) = \begin{cases} 1, & \text{si } \phi > 0, \\ 0, & \text{si } \phi = 0, \\ -1, & \text{si } \phi < 0. \end{cases} \quad (2.24)$$

2. Cette fonction signe peut être lissée proche de l'interface.

Afin d'alléger les notations, l'équation de redistanciation (2.25) peut s'écrire sous forme condensée

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi^0)G(\phi) = 0. \quad (2.25)$$

Remarque 1 L'équation (2.25) est une équation hyperbolique d'ordre 1 qui peut se réécrire sous la forme

$$\frac{\partial \phi}{\partial \tau} + \mathbf{w} \cdot \nabla \phi = \text{sign}(\phi^0), \quad (2.26)$$

avec $\mathbf{w} = \text{sign}(\phi^0)\mathbf{n}$ et $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$. L'équation (2.26) propage les caractéristiques suivant la normale à l'interface à la vitesse $\text{sign}(\phi^0) = \pm 1$ à partir de l'interface. En général, nous sommes intéressés à la propriété de distance uniquement dans un voisinage de l'interface, soit environ $3\Delta x$. Le temps fictif final peut alors être choisi $\tau_F = 3\Delta x$.

Remarque 2 L'équation de redistanciation (2.25) peut se réécrire sous la forme d'une équation de type Hamilton-Jacobi (Sussman *et al.*, 1994)

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi^0)\mathcal{H}(\phi, \nabla \phi) = 0, \quad (2.27)$$

où \mathcal{H} désigne l'Hamiltonien correspondant.

Pour la discrétisation en espace, nous allons brièvement introduire quelques méthodes, de la première introduite par Sussman *et al.* (1994), à la plus performante développé par Chéné *et al.* (2007) qui est une amélioration de la méthode proposé par Russo et Smereka (2000). Pour la discrétisation en temps, nous avons utilisé le schéma RK3 TVD (2.21) introduit par Gottlieb et Shu (1998). Cependant, pour des raisons de simplicité d'écriture, nous allons présenter les différentes méthodes de discrétisation en espace en couplant avec une méthode d'Euler explicite au premier ordre.

Discrétisation standard de type HJ-WENO

Afin d'alléger l'écriture, nous nous plaçons dans un espace en deux dimensions. Un schéma upwind de Godunov est utilisé pour discrétiser l'équation (2.27), soit :

$$\frac{\partial \phi}{\partial t} + \text{sign}(\phi^0)[\mathcal{H}_G(D_x^+ \phi, D_x^- \phi, D_y^+ \phi, D_y^- \phi)] = 0, \quad (2.28)$$

où $D_x^+ \phi, D_x^- \phi, D_y^+ \phi, D_y^- \phi$ correspondent aux dérivées de part et d'autre de l'interface (one-sided derivatives) et \mathcal{H}_G désigne une approximation numérique de l'Hamiltonien \mathcal{H} , *i.e.*

$$\mathcal{H}_G(a, b, c, d) = \begin{cases} \sqrt{\max(|a^+|^2, |b^-|^2) + \max(|c^+|^2, |d^-|^2)} - 1, & \text{si } \text{sign}(\phi^0) \leq 0, \\ \sqrt{\max(|a^-|^2, |b^+|^2) + \max(|c^-|^2, |d^+|^2)} - 1, & \text{si } \text{sign}(\phi^0) > 0, \end{cases} \quad (2.29)$$

avec $f^+ = \max(g, 0)$ et $f^- = \min(g, 0)$ pour tout f réel. La précision des méthodes de discrétisation dépend de la façon de calculer les dérivées $D_x^+ \phi, D_x^- \phi, D_y^+ \phi, D_y^- \phi$, notamment proche de l'interface. Pour la discrétisation standard, les dérivées D^\pm sont calculées en utilisant le schéma HJ-WENO de Jiang et Peng (2000), comme pour l'équation de transport. L'inconvénient majeur de ce schéma est de perturber quelque peu la position de l'interface. En effet, ce schéma n'est pas totalement upwind : le calcul des $\phi > 0$ utilise les informations des $\phi < 0$ proche de l'interface (et inversement). L'information ne se propage pas totalement à partir de l'interface dans la direction normale, mais elle peut localement autour de l'interface remonter en direction de celle-ci. Deux conséquences majeures en découlent :

- Une petite perturbation de l'interface produit de grosses erreurs sur le calcul des dérivées d'ordre élevées, comme par exemple la courbure (ordre 2) nécessaire pour imposer la tension de surface par exemple,
- Une perte de surface (ou volume en trois dimensions) est souvent présente, une goutte pouvant "disparaître".

La fonction $\text{sign}(\phi)$ est de plus lissée proche de l'interface pour limiter les erreurs liées à la discrétisation proche de l'interface. Ces problèmes empirent avec le nombre d'itérations utilisées pour faire converger la solution ϕ vers une fonction distance.

Schéma totalement upwind de Russo et Smereka (2000) : "The Subcell Fix"

Comme mentionné ci dessus, la forme de l'équation de redistanciation impose que l'information doit se propager de l'interface dans la direction normale. Partant de ce constat, Russo et Smereka (2000) ont proposé une solution numérique pour imposer cette condition. Afin d'empêcher l'utilisation de stencils de discrétisation avec des points de part et d'autre de l'interface (quand on est proche de celle-ci), ils proposent d'utiliser la valeur de ϕ à l'interface. La méthode proposée se faisant direction par direction, nous présentons le cas monodimensionnel. La méthode de Sussman *et al.* (1994) en 1D propose de résoudre :

$$\phi_i^{n+1} = \phi_i^n - \Delta t \text{sign} \phi_i^0 G(\phi)_i. \quad (2.30)$$

La fonction signe est donnée par (2.24) et

$$G(\phi)_i \phi_i = \begin{cases} \max(|a^+|, |b^-|) - 1, & \text{si } \phi_i^0 > 0, \\ \max(|a^+|, |b^-|) - 1, & \text{si } \phi_i^0 < 0, \end{cases} \quad (2.31)$$

avec $a = D_x^- \phi_i$ et $b = D_x^+ \phi_i$. Comme précédemment, on utilise la notation $f^+ = \max(f, 0)$ et $f^- = \min(f, 0)$ pour tout réel f .

Russo et Smereka (2000) propose de modifier le gradient, toujours upwind, proche de l'interface de la façon suivante :

$$D_x^{up} \phi_i = \begin{cases} \phi_i / D_i, & \text{si } \phi_i^0 \phi_{i-1}^0 < 0, \\ -\phi_i / D_i, & \text{si } \phi_i^0 \phi_{i+1}^0 < 0, \end{cases} \quad (2.32)$$

où

$$D_i = \Delta x \frac{\phi_i^0}{\Delta \phi_i^0} \quad (2.33)$$

est la distance signée à l'interface. Le terme $\Delta \phi_i^0$ peut être évalué de différentes manières, dont la plus stable numériquement est :

$$\Delta \phi_i^0 = \max \{ |\phi_{i+1}^0 - \phi_{i-1}^0| / 2, |\phi_{i+1}^0 - \phi_i^0|, |\phi_i^0 - \phi_{i-1}^0|, \epsilon \}. \quad (2.34)$$

Le paramètre ϵ est petit et empêche $\Delta \phi^0$ de s'annuler. La distance signée régularisée est alors :

$$S_i = \begin{cases} D_i / \Delta x, & \text{si } \phi_i^0 \phi_{i+1}^0 \leq 0 \quad \text{ou} \quad \phi_i^0 \phi_{i-1}^0 \leq 0, \\ \text{sign}(\phi_i^0), & \text{sinon,} \end{cases} \quad (2.35)$$

En injectant cette forme spécifique de la fonction signe dans (2.30) et en utilisant (2.31) on obtient :

$$\phi_i^{n+1} = \begin{cases} \phi_i^n + \frac{\Delta t}{\Delta x} (\text{sign}(\phi_i^0) |\phi_i^0| - D_i), & \text{si } \phi_i^0 \phi_{i+1}^0 < 0 \quad \text{or} \quad \phi_i^0 \phi_{i-1}^0 < 0, \\ \phi_i^n + \Delta t \text{sign}(\phi_i^0) G(\phi)_i, & \text{sinon.} \end{cases} \quad (2.36)$$

Nous avons utilisé le fait que $\text{sign}(D_i) = \text{sign}(\phi_i)$. Les gradients décentrés upwind sont pris au premier ordre, *i.e.*

$$(D^- \phi)_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} \quad \text{et} \quad (D^+ \phi)_i = \frac{\phi_{i+1} - \phi_i}{\Delta x}. \quad (2.37)$$

Ce schéma "Subcell Fix" ne modifie pas la position de l'interface, et la perte de volume (ou surface) devient indépendante du nombre d'itérations utilisées pour la redistanciation. Ce schéma 1D peut facilement être étendu au 2D ou 3D avec une précision à l'ordre 2 (Russo et Smereka, 2000). Pour obtenir l'ordre 2, les équations (2.30) et (2.31) sont résolues en remplaçant le calcul des dérivées (2.37) à l'ordre 1 par des approximations d'ordre 2. Cette idée utilise des ingrédients développés par Sussman et Fatemi (1999). On se donne 5 points autour du point x_i , soient $(x_p(k), f_p(k), k = -2, \dots, 2)$. Les dérivées à gauche et à droite, respectivement a et b sont calculées comme suit. On commence par calculer la table des différences divisées pour une interpolation de type Newton :

$$\begin{aligned} \Phi[k] &= f_p(k), & k &= -2, \dots, 2, \\ \Phi[k, k+1] &= \frac{\Phi[k+1] - \Phi[k]}{x_p(k+1) - x_p(k)}, & k &= -2, \dots, 1, \\ \Phi[k, k+2] &= \frac{\Phi[k+1, k+2] - \Phi[k, k+1]}{x_p(k+2) - x_p(k)}, & k &= -2, \dots, 0. \end{aligned} \quad (2.38)$$

On peut ensuite calculer a et b

$$\begin{aligned} a &= D_x^- \phi(x_p(0)) = \Phi[-1, 0] + c^-(x_p(0) - x_p(-1)), \\ b &= D_x^+ \phi(x_p(0)) = \Phi[0, 1] + c^+(x_p(0) - x_p(1)), \end{aligned} \quad (2.39)$$

avec

$$c^- = \minmod(\Phi[-2, 0], \Phi[-1, 1]), \quad c^+ = \minmod(\Phi[-1, 1], \Phi[0, 2]), \quad (2.40)$$

où la fonction \minmod est :

$$\minmod(\alpha, \beta) = \begin{cases} \alpha, & \text{si } |\alpha| \leq |\beta| \text{ et } \alpha\beta > 0, \\ \beta, & \text{si } |\alpha| > |\beta| \text{ et } \alpha\beta > 0, \\ 0, & \text{si } \alpha\beta \leq 0. \end{cases} \quad (2.41)$$

Les points du stencil de discrétisation sont alors choisis de la façon suivante. Si le point x_i n'est pas proche de l'interface, alors

$$x_p(x) = x_i(k), \quad f_p(k) = \phi_{i+k}, \quad k = -2, \dots, 2.$$

Si x_i est proche de moins d'une maille de l'interface, alors le point de l'interface remplace le point qui serait de l'autre côté de l'interface. Habituellement, le point de l'interface est calculé à l'aide d'une interpolation cubique de la level set initiale ϕ^0 en utilisant deux points de part et d'autre de l'interface. Une grande précision dans le calcul de la position est nécessaire puisque un point sur l'interface est maintenant utilisé pour calculer les dérivés décentrés proche de l'interface. Le schéma obtenu est alors d'ordre 2. Ce schéma fonctionnant direction par direction, l'extension au cas 2D et 3D est directe.

Amélioration du schéma "The Subcell Fix" par Chéné *et al.* (2007)

Le schéma Subcell fix peut être étendu à des ordres supérieurs en utilisant des reconstructions ENO. Ces schémas sont construits en ajoutant itérativement des nœuds afin de construire des interpolants d'ordre plus élevés sans recalculer les différences divisées déjà existantes. Cependant, comme mentionné par Chéné *et al.* (2007), deux conditions doivent être satisfaites pour un nœud proche de l'interface :

- Les nœuds à travers l'interface ne doivent pas être utilisés dans le calcul des interpolants pour conserver un schéma totalement upwind, à la manière de Russo et Smereka,
- Les nœuds à rajouter pour construire un interpolant d'ordre plus élevé doivent être choisis dans la région la plus lisse pour ϕ autour du point étudié.

Pour les points non adjacents à l'interface, un schéma standard HJ-WENO est utilisé.

Chéné *et al.* (2007) ont obtenu un schéma à l'ordre 4 pour ϕ , et donc à l'ordre 2 pour la courbure. Comme pour la méthode d'ordre 2 proposée par Russo et Smereka, une méthode de Newton basée sur les différences divisées est utilisée pour construire les interpolants. On se donne maintenant 7 points autour du point x_i , soient $(x_p(k), f_p(k), k = -3, \dots, 3)$.

En reprenant les notations de Russo et Smereka (par soucis d'homogénéité), les différences divisées sont alors :

$$\begin{aligned} \Phi[k] &= f_p(k), & k &= -3, \dots, 3, \\ \Phi[k, k+1] &= \frac{\Phi[k+1] - \Phi[k]}{x_p(k+1) - x_p(k)}, & k &= -3, \dots, 2, \\ \Phi[k, k+2] &= \frac{\Phi[k+1, k+2] - \Phi[k, k+1]}{x_p(k+2) - x_p(k)}, & k &= -3, \dots, 1, \\ \Phi[k, k+3] &= \frac{\Phi[k+1, k+3] - \Phi[k, k+2]}{x_p(k+3) - x_p(k)}, & k &= -3, \dots, 0. \end{aligned} \quad (2.42)$$

On peut ensuite calculer a et b

$$\begin{aligned} a &= \begin{cases} (x_p(0) - x_p(-1))(x_p(0) - x_p(-2))\text{MinAbs}(\Phi[-2, 1], \Phi[-3, 0]) & \text{si } |\Phi[-2, 0]| < |\Phi[-1, 1]|, \\ (x_p(0) - x_p(-1))(x_p(0) - x_p(1))\text{MinAbs}(\Phi[-2, 1], \Phi[-1, 2]) & \text{sinon .} \end{cases} \\ b &= \begin{cases} (x_p(0) - x_p(-1))(x_p(0) - x_p(1))\text{MinAbs}(\Phi[-2, 1], \Phi[-1, 2]) & \text{si } |\Phi[-1, 1]| < |\Phi[0, 2]|, \\ (x_p(0) - x_p(-1))(x_p(0) - x_p(2))\text{MinAbs}(\Phi[-1, 2], \Phi[0, 3]) & \text{sinon .} \end{cases} \end{aligned} \quad (2.43)$$

avec la fonction $MinAbs$ définie par :

$$MinAbs(\alpha, \beta) = \begin{cases} \alpha, & \text{si } |\alpha| < |\beta|, \\ \beta, & \text{sinon} \end{cases} \quad (2.44)$$

Les gradients sont alors calculés comme suit :

$$\begin{aligned} D_x^- \phi(x_p(0)) &= \Phi[-1, 0] + \minmod(\Phi[-2, 0], \Phi[-1, 1])(x_p(0) - x_p(-1)) + a, \\ D_x^+ \phi(x_p(0)) &= \Phi[0, 1] + \minmod(\Phi[-1, 1], \Phi[0, 2])(x_p(0) - x_p(1)) + b. \end{aligned} \quad (2.45)$$

La même procédure est ensuite appliquée aux autres directions spatiales. Cette amélioration du schéma de Russo et Smereka n'est pas anodine et permet d'obtenir une fonction distance à l'ordre 4 et donc une courbure à l'ordre 2.

Remarque sur le suivi lagrangien

Dans certaines applications, notamment les interfaces fluide/structures, les interfaces seront suivies à l'aide de marqueurs Lagrangiens. La position de l'interface pourra être reconstruite de façon précise avec des B-splines par exemple. Dans ces cas, la fonction distance sera calculée à chaque pas de temps de la façon suivante :

$$D(\mathbf{x}) = \min_{\mathbf{y} \in \Gamma} \|\mathbf{x} - \mathbf{y}\|_2 \text{sign}(\phi(\mathbf{x})). \quad (2.46)$$

Ce calcul peut s'avérer un peu lourd, mais permet d'avoir une grande précision sur la fonction distance, à condition que la distance entre deux marqueurs adjacents soit très petite par rapport au pas d'espace du maillage fluide.

2.3.3 Conservation de la masse

Les méthodes basées sur les fonctions level set ont longtemps souffert du problème de perte de masse. De nombreux chercheurs lui ont de ce fait préféré la méthode VOF qui assure cette conservation. Cependant, de nombreux progrès ont été effectués pour obtenir des fonctions level set qui conservent mieux la masse. Notamment, des méthodes numériques d'ordre élevées ont été développées pour assurer une meilleure conservation la masse, comme nous l'avons vu dans les paragraphes précédents. D'autres méthodes, avec des philosophies différentes, ont également été développées.

La première méthode est une redistanciation sous contraintes. Cette méthode, proposée par Sussman et Fatemi (1999), consiste à ajouter une contrainte de conservation de la masse lors de la résolution de l'équation de redistanciation. L'équation à résoudre est alors :

$$\phi_t + \text{sign}(\phi^0)(|\nabla\phi| - 1) = \lambda f(\phi), \quad (2.47)$$

en imposant que le volume contenu dans l'interface soit stationnaire,

$$\partial_t \int \Omega H(\phi) = 0.$$

Pour ne modifier que le volume dans un voisinage de l'interface Sussman et Fatemi choisissent $f(\phi) = H'(\phi)|\nabla\phi|$, ce qui amène à

$$\lambda = \frac{-\int_{\Omega} H'(\phi) \text{sign}(\phi^0)(1 - |\nabla\phi|)}{\int_{\Omega} H'(\phi) f(\phi)}.$$

D'un point de vue numérique le paramètre λ doit être évalué sur toutes les cellules et donc les intégrales sur tout l'espace Ω sont discréditées sur les mailles Ω_{ij} .

Une deuxième méthode consiste à coupler méthode VOF et level set. Nous savons que chacune des méthodes possède des avantages, mais également des inconvénients. Les avantages de l'une sont principalement les défauts de l'autre. En effet, quand la méthode VOF propose une bonne conservation de la masse et une mauvaise reconstruction géométrique de l'interface, la méthode level set propose une bonne reconstruction géométrique de l'interface et une mauvaise conservation de la masse. Il peut donc paraître astucieux de coupler la méthode VOF et la méthode Level Set. Sussman et Puckett (2000), puis Sussman (2003) ont proposé une méthode nommée CLSVOF qui permet via un algorithme complexe de coupler Level Set et VOF. Cette

méthode permet de réduire considérablement les défauts des deux méthodes, mais malheureusement produit également une petite perte des avantages.

Une troisième méthode, appelé le choix de contour peut être facilement appliquée (Couderc, 2007). L'idée de cette méthode est de considérer que n'importe quel contour défini par $\{\mathbf{x} \mid \phi(\mathbf{x}) = c\}$ peut représenter l'interface. Nous savons que des erreurs numériques ont été introduites après les étapes de transport et de redistanciation. La seule propriété qui est vérifiée est que la fonction obtenue, notée $\tilde{\phi}^n$, est bien une fonction distance. Par contre, il est probable qu'une petite perte de masse soit apparue. La méthode par choix de contours consiste à identifier l'interface au contour c de telle sorte que la masse soit préservée. En deux dimensions, nous calculons la surface initiale S_0 et la surface S après redistanciation à l'étape courante,

$$S_0 = \int_{\Omega} H^\epsilon(\phi^0) \quad \text{et} \quad S = \int_{\Omega} H^\epsilon(\tilde{\phi}^n), \quad (2.48)$$

où H^ϵ est un fonction Heaviside régularisée. Il est également nécessaire de calculer le périmètre à l'aide de la fonction Dirac régularisée γ^ϵ ,

$$P = \int_{\Omega} \gamma^\epsilon(\tilde{\phi}^n)$$

. Finalement, la level set est choisie

$$\phi^n = \tilde{\phi}^n - \frac{S - S_0}{P}.$$

Puisque cette étape est réalisée à chaque pas de temps, le contour choisit est $|c| \ll 1$, et donc la topologie de la fonction n'est que très peu affectée par ce choix. L'extension au cas tridimensionnels est immédiate.

Notons également qu'une nouvelle méthode nommée CLSMOF (Coupled Level set and moment of fluid) est apparue récemment (Jemison *et al.*, 2013).

2.4 Modélisation d'écoulements fluides en présence d'interfaces

Dans la suite de cette section nous allons présenter des méthodes de modélisation et de simulations numériques pour résoudre des écoulements de fluides incompressibles en présence d'interfaces. Nous allons distinguer deux types d'interfaces : les interfaces fluide/structure et les interfaces fluide/fluide. La combinaison des deux d'interfaces nous permettra d'appréhender des problèmes complexes d'interaction de deux phases liquides avec une structure.

De façon générale, nous considérons un domaine Ω qui se décompose en deux sous domaines Ω_f (pour le fluide) et Ω_s (pour le solide) séparés par une interface Γ_s qui est le zéro d'une fonction level set ψ_s . Par convention, nous choisissons $\psi_s > 0$ dans Ω_s et $\psi_s < 0$ dans Ω_f .

Le domaine de la phase fluide Ω_f peut se décomposer en deux sous domaines Ω_f^+ et Ω_f^- (eau et air par exemple) séparés par une interface Γ_f qui est le zéro d'une fonction level set ψ_f avec $\psi_f > 0$ dans Ω_f^+ et $\psi_f < 0$ dans Ω_f^- . Ceci peut naturellement s'étendre à plusieurs phases liquides.

Le domaine solide peut également se décomposer de différentes manières. Par exemple, si l'on considère N_s structures chacune définies avec une level-set ψ_{si} , $i = 1, \dots, N_s$ nous avons $\psi_s(\mathbf{x}) = \min_i \psi_{si}(\mathbf{x})$, pour tout points $\mathbf{x} \in \Omega$. De même, pour des matériaux composites, le domaine Ω_s peut se décomposer en sous domaines "par couches".

2.4.1 Modèle avec interfaces fluide/fluide/structure

Dans la suite, avec un souci de simplicité, nous nous limiterons à deux phases fluides avec une seule structure. La masse volumique de la structure est notée $\rho_s \in \mathbb{R}$. Les masses volumiques et viscosités dynamiques des fluides sont notées $\rho^\pm \in \mathbb{R}$ et $\mu^\pm \in \mathbb{R}$. Le champ de vitesse est $\mathbf{u} \in \mathbb{R}^d$ (d étant la dimension de l'espace étudié, généralement 2 ou 3), le champ de pression est $p \in \mathbb{R}$ et la gravité est $\mathbf{g} \in \mathbb{R}^d$ (que l'on peut prendre égale à zéro dans de nombreuses applications). De façon générale, en notant $D(\mathbf{u}) = \frac{\nabla \mathbf{u} + \nabla^T \mathbf{u}}{2}$, le système d'équations à résoudre est :

Equation de Navier-Stokes dans le domaine Ω_f :

$$\rho(\psi_f) \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu(\psi_f) D(\mathbf{u}) + \rho(\psi_f) \mathbf{g} \quad \text{dans } \Omega_f^+ \text{ et } \Omega_f^-, \quad (2.49a)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ dans } \Omega_f^+ \text{ et } \Omega_f^-, \quad (2.49b)$$

munies d'un jeu de conditions initiales ainsi que des conditions aux limites sur le bord du domaine $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_g(t)$ pour tout \mathbf{x} sur $\partial\Omega$ et sur les interfaces.

Conditions sur l'interface fluide/structure Γ_s : imperméabilité

$$\mathbf{u}(\mathbf{x}, t) = \hat{\mathbf{u}}(\mathbf{x}, t) \text{ sur } \Gamma_s. \quad (2.49c)$$

Le champ de vitesse $\hat{\mathbf{u}}$ peut être soit prescrit soit obtenu par un calcul des forces et moments exercés par le fluide sur l'obstacle via les lois de Newton. Nous reviendrons plus en détails sur point au chapitre suivant.

Conditions sur l'interface fluide/fluide Γ_f : Deux types de conditions de sauts.

Une condition cinématique qui se traduit par une discontinuité nulle de la vitesse à travers l'interface Γ_f ,

$$[\mathbf{u}] = 0 \text{ à travers } \Gamma_f, \quad (2.49d)$$

et une condition dynamique qui traduit l'équilibre des forces de pression, de frottement et de tension de surface³

$$[-p\mathbf{I} + 2\mu D(\mathbf{u})] \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \text{ à travers } \Gamma_f, \quad (2.49e)$$

où σ est le paramètre de tension de surface, κ la courbure à l'interface, et \mathbf{n} la normale unitaire à l'interface. Enfin, la masse volumique et la viscosité dynamique subissent un saut $[\rho] = \rho^+ - \rho^-$ et $[\mu] = \mu^+ - \mu^-$ à travers l'interface, et nous avons donc :

$$\rho(\psi_f) = \rho^+ + H(\psi_f)(\rho^- - \rho^+), \quad (2.49f)$$

$$\mu(\psi_f) = \mu^+ + H(\psi_f)(\mu^- - \mu^+). \quad (2.49g)$$

Transport des fonctions level-set : Enfin, les fonctions level-set satisfont des équations de transport (couplées éventuellement avec de la redistanciation). Classiquement, la fonction level set associée à l'interface fluide/fluide est transportée avec le champ de vitesse du fluide :

$$\frac{\partial \psi_f}{\partial t} + \mathbf{u} \cdot \nabla \psi_f = 0 \text{ dans } \Omega, \quad (2.49h)$$

Ensuite, la fonction level set associée à l'interface fluide/structure est transportée avec un champ de vitesse⁴ $\tilde{\mathbf{u}}$ qui satisfait $\tilde{\mathbf{u}} = \hat{\mathbf{u}}$ sur l'interface de la structure :

$$\frac{\partial \psi_s}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \psi_s = 0 \text{ dans } \Omega. \quad (2.49i)$$

Gestion de la ligne triple : Un modèle pour gérer les intersections des level set ψ_f et ψ_s doit être ajouté. En effet, si l'on considère par exemple un solide à vitesse nulle, la ligne d'eau $\psi_f = 0$ ne peut pas se déplacer sur $\psi_s = 0$ car elle satisfait l'équation (2.51g) avec $\mathbf{u} = \hat{\mathbf{u}} = 0$ sur Ω_s . Nous pouvons considérer un modèle de type Cahn-Hilliard ou un terme de dissipation peut être ajouté à l'équation (2.51g) . Un modèle de Cox peut également être utilisé.

Dans ce qui suit nous allons faire dégénérer le modèle (2.49) en étudiant le modèle mono fluide (masse volumique et viscosité constantes) avec une interface fluide/structure, et ensuite le modèle bifluide sans interface fluide/structure.

3. Nous avons supposé que la tension de surface est constante.

4. Ce champ est souvent confondu avec le champ de vitesse obtenu par intégration des équations de Navier-Stokes. Il peut également être obtenu par une méthode appelée *extension velocity* (Adalsteinsson et Sethian, 1999) qui peut permettre, sous certaines conditions, d'imposer directement que la fonction level set est une fonction distance signée à l'interface.

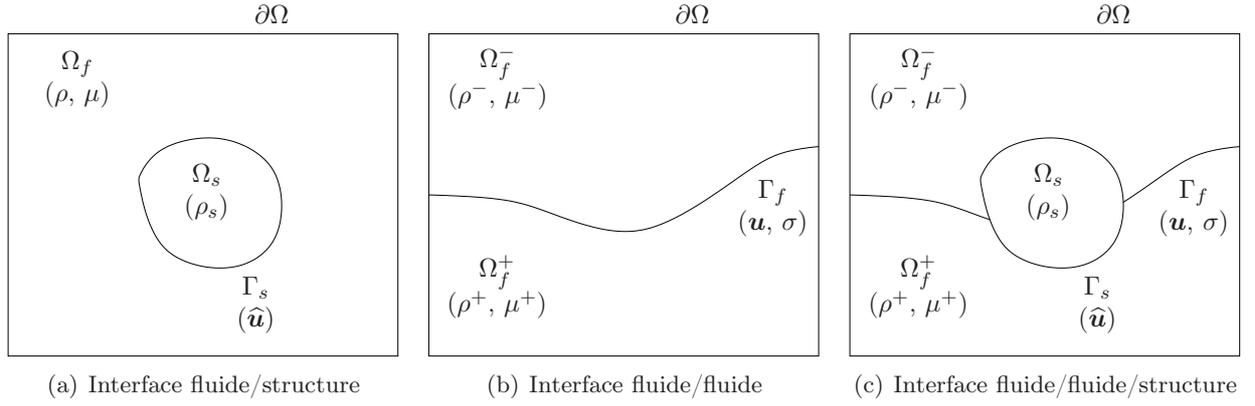


FIGURE 2.2 – Configurations des différents types d'écoulements étudiés.

2.4.2 Modèle avec interface fluide/structure

Equation de Navier-Stokes dans le domaine Ω_f :

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu D(\mathbf{u}) + \rho \mathbf{g} \text{ dans } \Omega_f, \quad (2.50a)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ dans } \Omega_f, \quad (2.50b)$$

+ conditions initiales + conditions aux limites sur le bord du domaine $\partial\Omega$.

Conditions sur l'interface fluide/structure Γ_s : imperméabilité

$$\mathbf{u}(\mathbf{x}, t) = \hat{\mathbf{u}}(\mathbf{x}, t) \text{ sur } \Gamma_s. \quad (2.50c)$$

Transport de la fonction leve-set :

$$\frac{\partial \psi_s}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \psi_s = 0 \text{ dans } \Omega. \quad (2.50d)$$

avec $\tilde{\mathbf{u}} = \mathbf{u}$ sur Γ_s .

2.4.3 Modèle avec interface fluide/fluide

Equation de Navier-Stokes dans le domaine Ω_f :

$$\rho(\psi_f) \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu(\psi_f) D(\mathbf{u}) + \rho(\psi_f) \mathbf{g} \text{ dans } \Omega_f^+ \text{ et } \Omega_f^-, \quad (2.51a)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ dans } \Omega_f^+ \text{ et } \Omega_f^-, \quad (2.51b)$$

+ conditions initiales + conditions aux limites sur le bord du domaine $\partial\Omega$.

Conditions sur l'interface fluide/fluide Γ_f :

$$[\mathbf{u}(\mathbf{x}, t)] = 0 \text{ à travers } \Gamma_f, \quad (2.51c)$$

$$[-pI + 2\mu D(\mathbf{u})] \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \text{ à travers } \Gamma_f, \quad (2.51d)$$

On a également :

$$\rho(\psi_f) = \rho^+ + H(\psi_f)(\rho^- - \rho^+), \quad (2.51e)$$

$$\mu(\psi_f) = \mu^+ + H(\psi_f)(\mu^- - \mu^+). \quad (2.51f)$$

Transport de la fonction leve set :

$$\frac{\partial \psi_f}{\partial t} + \mathbf{u} \cdot \nabla \psi_f = 0 \text{ dans } \Omega. \quad (2.51g)$$

2.5 Domaine fictif et interface fluide/structure

2.5.1 Configuration d'écoulement et modèle de Navier-Stokes incompressible

Dans toute cette étude nous nous sommes intéressés à des écoulements de fluides incompressibles. La configuration générique de notre étude est donnée par la figure 2.2(c). L'écoulement est gouverné par le système (2.49).

Une méthode pour résoudre numériquement le système (2.50) est d'imposer explicitement des conditions aux limites sur les interfaces fluide/structure. Numériquement, il peut sembler naturel d'utiliser un maillage conforme aux obstacles. Dans ce cas, l'interface est discrétisée et il est facile d'imposer des conditions aux limites. Ce problème se complique lorsque les interfaces se déplacent et surtout se déforment. Il serait alors nécessaire d'adapter le maillage à chaque instant.

Une alternative est alors de résoudre les équations précédentes sur un domaine $\tilde{\Omega}_f$ non adapté au domaine physique Ω_f avec idéalement⁵ $\Omega_f \subset \tilde{\Omega}_f$. Nous verrons dans la suite que la contrainte $\Omega_f \subset \tilde{\Omega}_f$ n'est pas toujours vérifiée, notamment pour la méthode Cut-Cells. Ce domaine peut être adapté au maillage fluide⁶ et peut donc être bien plus facile à gérer numériquement. Dans la suite nous allons utiliser un maillage fixe, cartésien. Idéalement, nous aimerions résoudre les équations de Navier-Stokes dans le domaine Ω_f

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu D(\mathbf{u}) + \rho \mathbf{g} \quad \text{dans } \tilde{\Omega}_f, \quad (2.52a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{dans } \tilde{\Omega}_f, \quad (2.52b)$$

et de modéliser implicitement la présence des interfaces. Les méthodes de modélisation et de résolution numérique d'EDP sur des domaines non adaptés aux domaines physiques sont appelées méthodes de domaines fictifs. Ces méthodes ont été introduites par Saul'ev (1963). Dans ce qui suit nous allons présenter deux types de méthodes de domaines fictifs : les méthodes continues et les méthode discrètes. Deux ingrédients entrent en jeu, le modèle et les schémas de discrétisation du modèle. Pour l'approche continue, le modèle est modifié pour prendre en compte l'interface et les schémas numériques sont les mêmes dans tout le domaine de calcul. Pour l'approche discrète, le modèle est inchangé, mais les schémas numériques sont adaptés pour prendre en compte les interfaces d'un point de vue discret. Nous verrons qu'il est également possible de coupler approche continue et approche discrète. Deux types de modélisation d'interface sont utilisés : interface diffuse (interface lissée sur une ou plusieurs cellules) et interface fine (traitée se façon "raide"). Nous venons de présenter à la section précédente des méthodes pour suivre et capturer les interfaces. A partir de ces interfaces, nous devons modéliser, éventuellement de façon numérique, la présence de l'obstacle. L'obstacle peut alors être considéré comme étant une phase.

2.5.2 Approche continue

Dans ce type d'approche nous modifions les équations et conservons les mêmes schémas numériques⁷ de résolution dans tout le domaine. Nous introduisons alors une force extérieure $\mathbf{f} \in \mathbb{R}^d$ qui modélise la présence d'une éventuelle interface, appelée frontière immergée. Les schémas numériques de discrétisation des équations ne sont pas impactés par la présence de l'interface. En général, le domaine $\tilde{\Omega}_f$ peut être confondu avec le domaine tout entier Ω , et le modèle à résoudre est alors :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot 2\mu D(\mathbf{u}) + \mathbf{g} + \frac{\mathbf{f}}{\rho} \quad \text{dans } \Omega, \quad (2.53a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{dans } \Omega, \quad (2.53b)$$

5. Le domaine tout entier Ω est un bon candidat.

6. Un domaine $\tilde{\Omega}_f$ en marche d'escalier est également envisageable.

7. Dans cette étude nous utilisons des schémas aux différences finies. Il est bien sûr possible d'utiliser de volumes finis, et même des éléments finis sur des maillages non structurés fixes dans l'espace (Abgrall *et al.*, 2014).

munies d'une condition initiale et de conditions aux limites sur les bords extérieurs. Dans le cas d'un écoulement incompressible avec $\rho = Cste$ et $\mu = Cste$, et donc $\nu = \mu/\rho = Cste$, nous avons $\frac{1}{\rho}\nabla \cdot 2\mu D(\mathbf{u}) = \nu\Delta\mathbf{u}$.

Nous remarquons que nous n'imposons pas explicitement de conditions aux limites sur les interfaces fluide/solide. En effet, les forces extérieures \mathbf{f} sont imposées pour forcer le champ de vitesse à satisfaire les bonnes conditions aux limites sur ces interfaces.

La première méthode de frontière immergée a été introduite par Peskin en 1972 pour modéliser et étudier les écoulements sanguins proche des valves cardiaques. Cette méthode ayant été largement étudiée dans la littérature, nous ne le présenterons pas dans ce mémoire.

La seconde classe de méthodes est basée sur la pénalisation des équations de quantité de mouvement. Là encore, plusieurs méthodes ont été développées, comme la méthode de pénalisation pour des obstacles qui peuvent bouger d'une manière prescrite (Fujita et Sauer, 1969), la pénalisation de Darcy introduite par Arquís (1984) ou encore la méthode de pénalisation du tenseur des contraintes utilisée par Glowinski *et al.* (2001). Pour la plupart, ces méthodes sont principalement utilisées pour des obstacles fixes ou rigides. La méthode de pénalisation dans le cas des solides rigides dont le déplacement n'est pas connu a été introduite par Conca *et al.* (2000) (pénalisation L_2) et par San Martín et Starovoitov (2002) (pénalisation H_1). Enfin, la méthode de pénalisation volumique est la suivante :

$$\mathbf{f} = \frac{\chi}{K}(\hat{\mathbf{u}} - \mathbf{u}), \quad (2.53c)$$

où χ est une fonction indicatrice ($\chi = 1$ dans l'obstacle et $\chi = 0$ dans les fluides), K est un paramètre de perméabilité choisi très petit de tel sorte que le solide soit considéré comme étant imperméable, et enfin $\hat{\mathbf{u}}$ est la vitesse du corps solide. Cette vitesse peut dépendre du temps et de l'espace. Une justification complète couplée à des estimations d'erreurs liées à cette méthode ont été données par Angot *et al.* (1999). Un point important est que le champ de pression est correct à travers les interfaces⁸. Nous avons utilisé cette méthode pour modéliser et simuler le mouvement autopropulsé de corps déformables (Bergmann et Iollo, 2011). L'avantage de cette méthode est principalement sa simplicité de mise en œuvre : en effet, les mêmes équations (2.53b) et (2.53a) couplées à (2.53c) sont résolues dans tout le domaine. Son principal défaut est par contre l'ordre de précision qui ne dépasse par l'unité si on ne prend pas de précautions particulières.

2.5.3 Approche discrète

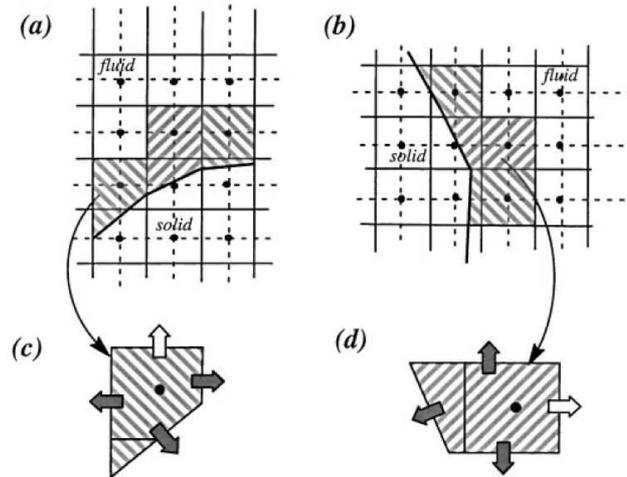
Une autre solution est de ne pas modifier les équations (utiliser $\mathbf{f} = 0$) mais de modifier les schémas numériques proche des interfaces. Plusieurs choix pour le domaine $\tilde{\Omega}_f$ sont possibles. Le premier consiste à couper les cellules en présence d'interface. Une approximation linéaire est utilisée : les cellules sont coupées par une droite en 2D et par un plan en 3D. On parle alors de méthode de cut-cells (Clarke *et al.*, 1986). Sans rentrer dans des détails, cette méthode couplée à un schéma volumes finis permet une formulation totalement conservative (Cheny et Botella, 2010). Cette méthode est cependant très difficile à mettre en œuvre numériquement car il existe plus d'une centaine de configurations pour les cut-cells en 3D. Il n'existe à ce jour que très peu de codes de calcul 3D cut-cells (Mittal et Iaccarino, 2005). La figure 2.5.3, tirée de Ye *et al.* (1999), présente la méthode Cut-Cells en 2D.

Les méthodes de frontières immergées discrètes utilisent généralement un domaine minimal $\tilde{\Omega}_f$ qui peut être choisi conforme au maillage cartésien en étendant le domaine Ω_f sur quelques mailles⁹ à l'intérieur des domaines "solides" Ω_s . Eventuellement, le domaine tout entier Ω peut être considéré. Plusieurs types de méthodes existent et le lecteur intéressé pourra se référer à la synthèse¹⁰ de Mittal et Iaccarino (2005). Dans la thèse de Hovnarian (2012) nous avons fait une synthèse plus détaillée des différentes approches. Nous allons présenter la base des méthodes des cellules fantômes (*Ghost Cell Method*, GCM) que nous avons utilisées dans cette étude. Cette méthode est une variante de la méthode Ghost Fluid introduite par Fedkiw *et al.* (1999a). La méthode Ghost Fluid a été utilisée avec succès dans de nombreuses application comme la résolution d'une équation de Poisson sur un domaine irrégulier (Gibou *et al.*, 2002). Sans rentrer dans les détails, cette méthode peut devenir instable pour des points très proches de l'interface, et il convient alors de déplacer légèrement l'interface pour que cette dernière soit confondue avec le point de maillage cartésien

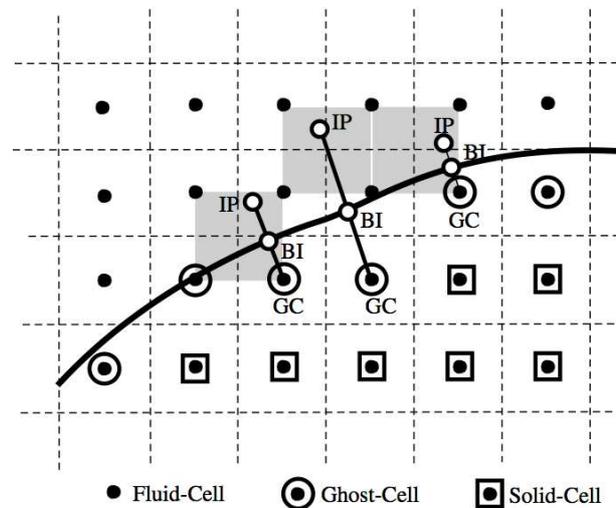
8. Ce point est important lorsque les équations de Navier-Stokes sont résolues en temps par des méthodes de projection à pas fractionnaire. Il faut généralement imposer des conditions aux limites pour la correction de pression sur les interfaces. La pénalisation volumique permet d'éviter cette étape, qui peut être coûteuse car l'opérateur doit être modifié et un nouveau préconditionneur doit être recalculé à chaque itération.

9. Le nombre de mailles dépend de la précision du schéma de discrétisation en espace choisi.

10. Cette synthèse devrait être mise à jour en raison des importants développements effectués ces dernières années.

FIGURE 2.3 – Schématisation de la méthode Cut-Cells en 2D par Ye *et al.* (1999).

considéré. Pour remédier à ce (léger) problème, les méthodes GCM ont été développées. Le principe de ces méthodes est représenté sur la figure 2.5.3 pour un cas 2D, tirée de Ghias *et al.* (2007). Le but est de modifier

FIGURE 2.4 – Schématisation de la méthode GCM de Ghias *et al.* (2007)

la valeur de la vitesse \mathbf{u}_{GC} sur les cellules fantômes (GC, *ghost cells*) afin de satisfaire les bonnes conditions aux limites \mathbf{u}_{BI} sur l'interface au points BI. Pour ce faire, nous utilisons la vitesse \mathbf{u}_{IP} du point symétrique (*Image Point*). La position du point \mathbf{x}_{IP} peut facilement être calculée à partir de la fonction distance signée ϕ par $\mathbf{x}_{IP} = \mathbf{x}_{GC} + \phi \nabla \phi$, de même la position du point symétrique est donnée par $\mathbf{x}_{IP} = \mathbf{x}_{GC} + 2\phi \nabla \phi$. Nous pouvons alors imposer $\mathbf{u}_{GC} = 2\mathbf{u}_{BI} - \mathbf{u}_{IP}$. La valeur \mathbf{u}_{BI} est connue¹¹ et la valeur \mathbf{u}_{IP} doit être interpolée en utilisant les informations de ses voisins proches. Par exemple, pour un cas 3D, nous identifions la cellule "cubique" à laquelle appartient le point IP . La valeur de la vitesse est alors obtenue à l'aide d'une interpolation faisant intervenir les valeurs de vitesse aux 8 sommets du cube. Pour chaque composante de vitesse, notée ξ nous connaissons les valeurs $\{\xi\}^T = \{\xi_1, \xi_2, \dots, \xi_8\}$ et pouvons écrire

$$\xi(x_1, x_2, x_3) = C_1 x_1 x_2 x_3 + C_2 x_1 x_2 + C_3 x_2 x_3 + C_4 x_1 x_3 + C_5 x_1 + C_6 x_3 + C_7 x_3 + C_8.$$

11. Cette valeur est soit fixée par l'utilisateur pour un mouvement imposé du corps, soit découle d'un calcul d'interaction fluide/structure (one-way ou two-way) pour un mouvement libre ou autopropulsé.

Les coefficients $\{C\}$ peuvent facilement être calculés par $\{C\} = \{V\}^{-1}\{\xi\}$, où la Matrice de Vandermonde est

$$V = \begin{pmatrix} x_1x_2x_3|_1 & x_1x_2|_1 & x_2x_3|_1 & x_1x_3|_1 & x_1|_1 & x_2|_1 & x_3|_1 & 1 \\ x_1x_2x_3|_2 & x_1x_2|_2 & x_2x_3|_2 & x_1x_3|_2 & x_1|_2 & x_2|_2 & x_3|_2 & 1 \\ \vdots & \vdots \\ x_1x_2x_3|_8 & x_1x_2|_8 & x_2x_3|_8 & x_1x_3|_8 & x_1|_8 & x_2|_8 & x_3|_8 & 1 \end{pmatrix}.$$

Chaque évaluation de \mathbf{u}_{GC} ne nécessite alors que la résolution d'un système linéaire 8×8 en 3D et 4×4 en 2D. Cette méthode permet d'obtenir l'ordre 2 numériquement¹² (Ghias *et al.*, 2007; Mittal *et al.*, 2008). Notons enfin que, le cas échéant, le point à modifier n'est pas considéré dans les 8 points, mais est remplacé par son projeté sur l'interface.

Cette méthode ne permet pas d'obtenir un champ de pression correct proche des interfaces sans imposer $\frac{\partial p}{\partial \mathbf{n}} = 0$ par exemple. Dans le cas d'une méthode de projection à pas fractionnaires, il est alors nécessaire d'imposer des conditions aux limites à l'incrément de pression.

2.5.4 Une méthode hybride au second ordre

Cette méthode a en partie été développée dans le cadre de la thèse de Jessica Hovnanian (2012). L'idée est de coupler les avantages des méthodes de frontières immergées (précision d'ordre 2 sur la frontière fluide/structure) et les avantages de la méthode de pénalisation volumique (modèle physique et simplicité de mise en œuvre, non imposition explicite de condition de pression sur les interfaces fluide/structure) tout en ignorant leurs défauts mutuels (ordre 1 pour pénalisation et imposition de condition de pression pour frontières immergées). Le modèle *continu* à résoudre est alors :

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot 2\mu D(\mathbf{u}) + \mathbf{g} + \frac{\chi}{K} (\hat{\mathbf{u}}_2 - \mathbf{u}) \quad \text{dans } \Omega, \quad (2.54a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{dans } \Omega, \quad (2.54b)$$

où la valeur de la vitesse de pénalisation $\hat{\mathbf{u}}_2$ est calculée d'un point de vue *discret* de façon à obtenir la valeur désirée sur la frontière de l'obstacle. La vitesse en chaque point du solide sera notée $\hat{\mathbf{u}}_1$. Les vitesses $\hat{\mathbf{u}}_2$ et $\hat{\mathbf{u}}_1$ diffèrent uniquement sur les mailles ayant au moins un voisin¹³ dans le fluide.

2.5.5 Résolutions numériques des équations

Discrétisation en espace

Dans cette étude nous avons choisi d'utiliser une discrétisation des équations de Navier-Stokes par différences finies. Ce choix se justifie par la mise en œuvre de schémas numériques très simples et par le fait que nous utilisons des grilles cartésiennes uniformes avec lesquelles les différences finies sont confondues avec des volumes finis. Un regain d'intérêt pour les schémas aux différences finies est apparu. En effet, les différences finies sont de plus en plus utilisées, même pour traiter des applications complexes comme des écoulements compressibles en présence de chocs (Cinnella, 2013) et des problèmes en aéroacoustique (Marsden, 2013).

Les approches de résolution sur maillage cartésien se divisent en deux groupes : l'utilisation de grilles collocalisées et les grilles MAC (Marker And Cells). Ces grilles sont représentées sur la figure 2.5.5, où la grille est collocalisée pour les variables (u, v, p) et MAC pour les variables (U, V, p) .

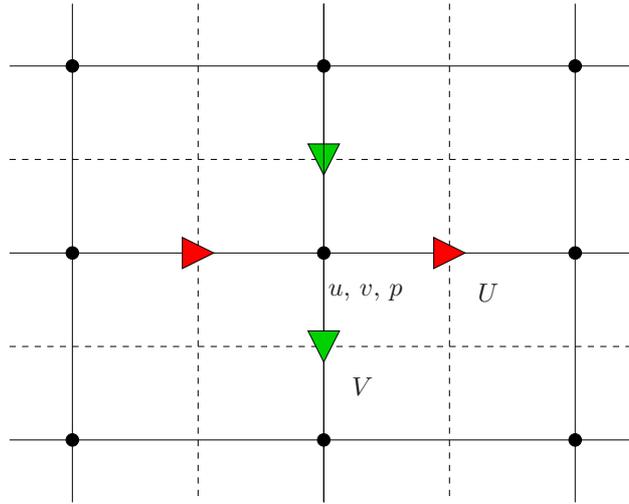
L'avantage d'une grille collocalisée est que toutes les variables sont placées au même endroit et il est donc aisé d'imposer des conditions aux limites sur toutes les composantes du vecteur vitesse. L'inconvénient est qu'il est possible de voir apparaître des modes parasites. En effet, la divergence n'est pas satisfaite d'un point de vue discret (simplement à la précision du schéma numérique) et nous pouvons observer une oscillation de pression sur $2\Delta x$.

L'avantage d'une grille MAC est que la divergence peut être satisfaite à la précision machine. Par contre, il peut paraître plus difficile d'imposer des conditions aux limites, en particulier pour les méthodes de pénalisation ou de Ghost Cells.

Dans ce qui suit, nous allons utiliser une grille hybride qui permet de tirer avantage des deux types de grilles au prix du stockage d'un champ de vitesse supplémentaire. Cette méthode a été motivée par de

12. L'ordre 2 est uniquement obtenu si les schémas de discrétisation des équations de Navier-Stokes sont d'ordre 2 ou plus.

13. On appelle voisin d'un point (x, y, z) tout point $(x \pm \Delta x, y, z)$, $(x, y \pm \Delta y, z)$ ou $(x, y, z \pm \Delta z)$.

FIGURE 2.5 – Grille collocalisée pour (u, v, p) et MAC pour (U, V, p) .

nombreuses discussions avec Rajat Mittal lors de son séjour à Bordeaux. Cette méthode est également basée sur les idées initiales de Rhie et Chow (1983) où les vitesses aux faces sont obtenues par interpolations des vitesses aux centres des cellules.

Nous avons discrétisé tous les termes des équations avec des différences finies centrées au second ordre. Pour certaines applications où la convection est dominante, nous avons utilisé des schémas upwind au troisième ordre, voir un schéma WENO5, pour discrétiser les termes convectifs.

Discrétisation en temps

Il existe différentes méthodes de résolution en temps des équations de Navier-Stokes incompressibles. La première est une méthode de lagrangien augmenté basée sur un algorithme d'Uzawa. Cette méthode est très performante pour des écoulements bas Reynolds, voir Stokes. Par contre la convergence est bien plus délicate pour des nombres de Reynolds modérés à forts. Un autre type de méthode est basée sur des schémas de projection. Les plus connues sont les méthodes à correction de pression, initialement introduite par Chorin (1968) et Temam (1969). Notons qu'il existe également des méthodes à correction de vitesse. Ces schémas de projection sont eux plus adaptés pour traiter des nombres de Reynolds élevés. Dans ce qui suit nous allons utiliser une méthode à pas fractionnaire basée sur une correction de pression. Puisque nous souhaitons coupler pénalisation volumique et GCM discrète, nous prenons $\tilde{\Omega}_f = \Omega$. Nous devons donc résoudre les équations de conservation de la masse (2.54b) et de conservation de la quantité de mouvement (2.54a). Une discrétisation au second ordre de l'équation de quantité de mouvement est utilisée. Dans ce qui suit, pour simplifier les notations, nous posons $L(\rho, \mu, \mathbf{u}) = \frac{1}{\rho} \nabla \cdot 2\mu D(\mathbf{u})$.

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+\frac{1}{2}} = -\frac{1}{\rho} \nabla p^{n+\frac{1}{2}} + L(\rho, \mu, \mathbf{u})^{n+\frac{1}{2}} + \left[\frac{\chi}{K} (\hat{\mathbf{u}}_2 - \mathbf{u}) \right]^{n+\frac{1}{2}}, \quad (2.55a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (2.55b)$$

avec les conditions aux limites sur les bords extérieurs $\partial\Omega$ du domaine

$$\mathbf{u}^{n+1} = \mathbf{u}_b^{n+1}. \quad (2.55c)$$

Puisque nous allons utiliser une méthode à pas fractionnaires, nous allons séparer certains termes de l'équation (2.55a) et les réécrire de façon différentes.

– La dérivée temporelle est :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \underbrace{\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t}}_{III} + \underbrace{\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^*}{\Delta t}}_{II} + \underbrace{\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t}}_I, \quad (2.56)$$

– le terme de convection est obtenu avec une extrapolation au second ordre :

$$[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+\frac{1}{2}} = \underbrace{\frac{3}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^n - \frac{1}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n-1}}_I, \quad (2.57)$$

– le terme de pression est :

$$-\frac{1}{\rho} \nabla p^{n+\frac{1}{2}} = \underbrace{-\frac{1}{\rho} \nabla q}_I + \underbrace{\frac{1}{\rho} \nabla (q - p^{n+\frac{1}{2}})}_{II}, \quad (2.58)$$

– le terme de dissipation est obtenu avec une interpolation au second ordre de type Crank-Nicholson :

$$\begin{aligned} L(\rho, \mu, \mathbf{u})^{n+\frac{1}{2}} &= \frac{1}{2} (L(\rho, \mu, \mathbf{u})^{n+1} + L(\rho, \mu, \mathbf{u})^n) \\ &= \underbrace{\frac{1}{2} (L(\rho, \mu, \mathbf{u})^* + L(\rho, \mu, \mathbf{u})^n)}_I + \underbrace{\frac{1}{2} (L(\rho, \mu, \mathbf{u})^{n+1} - L(\rho, \mu, \mathbf{u})^*)}_{III} \end{aligned} \quad (2.59)$$

– le terme de pénalisation est ¹⁴ :

$$\left[\frac{\chi}{K} (\hat{\mathbf{u}}_2 - \mathbf{u}) \right]^{n+\frac{1}{2}} = \underbrace{\frac{1}{2K} \chi^{n+1} (\hat{\mathbf{u}}_2^{n+1} - \mathbf{u}^{n+1})}_{III} + \underbrace{\frac{1}{2K} \chi^n (\hat{\mathbf{u}}_2^n - \mathbf{u}^n)}_{III} + \underbrace{\frac{1}{K} \chi^n (\hat{\mathbf{u}}_1^n - \mathbf{u}^*)}_I - \underbrace{\frac{1}{K} \chi^n (\hat{\mathbf{u}}_1^n - \mathbf{u}^*)}_{III} \quad (2.60)$$

où $\hat{\mathbf{u}}_1$ et $\hat{\mathbf{u}}_2$ sont respectivement les vitesses de pénalisation à l'ordre 1 et pénalisation à l'ordre 2. Ces vitesses de pénalisation sont identiques, excepté sur les premières mailles dans le solide où la vitesse au second ordre $\hat{\mathbf{u}}_2$ est calculée par une méthode GCM.

Les termes *I* correspondent à la prédiction du champ de vitesse avec pénalisation d'ordre 1. Les termes *II* correspondent à la correction des champs pour satisfaire la divergence nulle, et enfin les termes *III* correspondent à la correction avec la méthode IPC d'ordre 2 (frontières immergées) plus le reste des termes.

L'algorithme de résolution est le suivant :

Etape I : prédiction on résout l'équation qui rassemble les termes *I* (à partir d'un champ de pression q)

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^n - \frac{1}{2} [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n-1} = -\frac{1}{\rho} \nabla q + \frac{1}{2} (L(\rho, \mu, \mathbf{u})^* + L(\rho, \mu, \mathbf{u})^n) + \lambda \chi^n (\hat{\mathbf{u}}_1^n - \mathbf{u}^*), \quad (2.61a)$$

avec les conditions aux limites

$$B(\mathbf{u}^*) = 0. \quad (2.61b)$$

Le champ de vitesse obtenu par prédiction \mathbf{u}^* n'est *a priori* pas à divergence nulle, $\nabla \cdot \mathbf{u}^* \neq 0$, mais $\mathbf{u}^* \xrightarrow{\lambda \rightarrow +\infty} \mathbf{u}_1^n$ dans l'obstacle (à l'ordre 1 en espace). A ce stade, deux choix sont généralement utilisés pour la pression prédite q .

Le premier choix est $q = 0$. Dans ce cas, le champ prédit peut être relativement différent du champ souhaité \mathbf{u}^{n+1} et l'imposition de la condition limite $B(\mathbf{u}^*)$ demande quelques ajustements. En effet, il manque le gradient de pression à ce champ de vitesse, et il conviendrait alors d'inclure une estimation non triviale de ∇p^{n+1} dans $B(\mathbf{u}^*)$.

Le second choix, proposé Goda (1979), consiste à poser $q = p^{n-\frac{1}{2}}$, ou alors une extrapolation d'ordre supérieure pour approcher $p^{n+\frac{1}{2}}$. Cette méthode est habituellement appelée méthode incrémentale. Dans ce cas, q est une relativement bonne approximation de $p^{n+\frac{1}{2}}$ et il semble plus naturel d'imposer pour \mathbf{u}^* les mêmes conditions aux limites que pour \mathbf{u}^{n+1} , *i.e.* $B(\mathbf{u}^*) = (\mathbf{u}^* - \mathbf{u}_b)|_{\partial\Omega} = 0$.

14. Le choix de faire des moyennes sur la fonction caractéristique χ est le plus simple, même s'il n'est pas le plus réaliste.

Etape II : Correction on calcule la divergence de l'équation qui rassemble les termes II, en imposant $\nabla \cdot \tilde{\mathbf{u}}^{n+1} = 0$

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \psi \right) = -\frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \quad \text{avec } \psi = p^{n+\frac{1}{2}} - q. \quad (2.62a)$$

avec les conditions aux limites

$$\frac{\partial \psi}{\partial \mathbf{n}} = 0. \quad (2.62b)$$

Dans le cas ou la masse volumique est constante dans tout le domaine, nous pouvons poser $\psi = \frac{p^{n+\frac{1}{2}} - q}{\rho}$ et résoudre $\Delta \psi = -\frac{\nabla \cdot \mathbf{u}^*}{\Delta t}$.

Afin que ce problème soit bien posé, on fixe également la valeur de la pression, par exemple sur un point de la frontière amont. Le champ \mathbf{u}^* étant évalué avec pénalisation, il n'est pas nécessaire d'imposer explicitement des conditions aux limites sur l'interface pour la pression (Angot *et al.*, 1999). On corrige ensuite les champs de vitesse et de pression :

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{u}^* - \Delta t \frac{\nabla \psi}{\rho}, \quad (2.63a)$$

$$p^{n+\frac{1}{2}} = q + \psi. \quad (2.63b)$$

La correction de pression proposée ici est la plus utilisée. Cependant, Brown *et al.* (2001) proposent une correction qui permet de retrouver l'ordre 2 sur la pression jusqu'aux bords du domaine, soit $p^{n+\frac{1}{2}} = q + \psi + \frac{\Delta t}{2} \frac{\rho}{\mu} \Delta \psi$. Timmermans *et al.* (1996) proposent quant à eux un schéma rotationnel $p^{n+\frac{1}{2}} = q + \psi - \frac{1}{2} \frac{\rho}{\mu} \nabla \cdot \mathbf{u}^*$. D'autres choix sont encore possible, mais pratiquement, la correction (2.63b) donne de très bons résultats pour des nombres de Reynolds modérés.

Etape III : Transport et correction IPC au second ordre Si nécessaire, si $\chi^{n+1} \neq \chi^n$, le transport de la fonction caractéristique (et distance signée) est effectué pour obtenir χ^{n+1} , et on résout l'équation composée des termes III :

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} &= \frac{1}{2} (L(\rho, \mu, \mathbf{u})^{n+1} - L(\rho, \mu, \mathbf{u})^*) \\ &+ \frac{\lambda}{2} \chi^{n+1} (\hat{\mathbf{u}}_2^{n+1} - \mathbf{u}^{n+1}) + \frac{\lambda}{2} \chi^n (\hat{\mathbf{u}}_2^n - \mathbf{u}^n) - \lambda \chi^n (\hat{\mathbf{u}}_1^n - \mathbf{u}^*) \end{aligned} \quad (2.64)$$

Les deux derniers termes tendent vers zéro car :

$$\mathbf{u}^* \xrightarrow{\lambda \rightarrow +\infty} \mathbf{u}_1^n \quad \text{pour } \chi^n = 1 \quad (\text{imposé dans l'étape de prédiction (2.61a)})$$

$$\mathbf{u}^n \xrightarrow{\lambda \rightarrow +\infty} \mathbf{u}_2^n \quad \text{pour } \chi^n = 1 \quad (\text{imposé au pas de temps précédent})$$

De plus, on néglige souvent le terme de correction de la dissipation $\frac{1}{2} (L(\rho, \mu, \mathbf{u})^{n+1} - L(\rho, \mu, \mathbf{u})^*)$, et l'équation à résoudre est simplement :

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} = \frac{\lambda}{2} \chi^{n+1} (\hat{\mathbf{u}}_2^{n+1} - \mathbf{u}^{n+1}). \quad (2.65)$$

Ce schéma numérique est particulièrement adapté pour calculer des écoulements en présence d'interfaces fluides/structures. Nous allons traiter dans la section suivante le cas des interfaces fluide/fluide.

Conditions de stabilité sur le pas de temps : afin d'obtenir un algorithme stable numériquement nous devons imposer des conditions sur le pas de temps Δt . Le système est gouverné par des termes de convection et des termes de diffusion visqueuse.

La première condition, connue sous le nom de condition CFL (*Courant, Friedrichs et Lewy*), impose qu'une particule fluide ne doit d'une distance excédant la taille de la maille dans chaque direction spatiale. Cette condition impose donc :

$$\Delta t \underbrace{\left(\max \left(\frac{\|u\|_\infty}{\Delta x}, \frac{\|v\|_\infty}{\Delta y}, \frac{\|w\|_\infty}{\Delta z} \right) \right)}_{C_c} \leq 1.$$

Si les termes visqueux sont traités de manière explicite dans l'étape de prédiction (2.61a), une condition de stabilité supplémentaire est nécessaire. Cette condition est :

$$\Delta t \underbrace{\left(\frac{\mu}{\rho} \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right) \right)}_{C_\mu} \leq 1.$$

Le cas échéant, une contrainte liée aux forces de volume $\mathbf{g} = (g_x, g_y, g_z)$ peut être ajoutée. A chaque pas de temps, une approximation linéaire de la vitesse suivant x est $\|u\|_\infty + \Delta t \|g_x\|_\infty$, et la condition CFL devient pour la direction x :

$$\Delta t \left(\frac{\|u\|_\infty}{\Delta x} + \sqrt{\left(\frac{\|u\|_\infty}{\Delta x} \right)^2 + \frac{4\|g_x\|_\infty}{\Delta x}} \right) \leq 2.$$

On procède de même pour les autres directions spatiales et on obtient finalement :

$$\Delta t \left(C_c + C_\nu + \sqrt{(C_c + C_\nu)^2 + 4C_g} \right) \leq 2, \quad (2.66)$$

avec $C_g = \max \left(\frac{\|g_x\|_\infty}{\Delta x}, \frac{\|g_y\|_\infty}{\Delta y}, \frac{\|g_z\|_\infty}{\Delta z} \right)$.

2.6 Modélisation et résolution numérique des interfaces fluide/fluide

Les écoulements en présence d'interfaces fluide/fluide sont régis numériquement par le modèle (2.51).

En particulier, on rappelle que l'on a un saut de masse volumique et de viscosité dynamique à travers l'interface fluide/fluide, à savoir :

$$\rho = \rho^+ + H(\phi)(\rho^- - \rho^+), \quad (2.67)$$

$$\mu = \mu^+ + H(\phi)(\mu^- - \mu^+). \quad (2.68)$$

Dans ce qui suit nous allons présenter deux méthodes pour traiter les conditions de sauts à l'interface : la méthode CSF (Continuum Surface Force) dans laquelle nous utiliserons une interface lissée (la fonction de Heaviside H est lissée) et la méthode GFM (Ghost Fluid Method) dans laquelle une interface raide sera utilisée (H est raide).

2.6.1 Interface lisse et *Continuum Surface Force* (CSF)

Méthode CSF (Continuum Surface Force) a été initialement introduite par Brackbill *et al.* (1992) et développée par Sussman *et al.* (1994). L'idée de la méthode CSF est de traiter le terme de tension de surface comme une force volumique supplémentaire dans les équations de la conservation de la quantité de mouvement. Le saut

$$[-pI + 2\mu D(\mathbf{u})] \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \text{ à travers } \Gamma_f, \quad (2.69)$$

est considéré sous une forme intégrale et nous avons alors :

$$\rho(\psi_f) \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu(\psi_f) D(\mathbf{u}) + \sigma \kappa \delta(\psi_f) \mathbf{n} + \rho(\psi_f) \mathbf{g} \text{ dans } \Omega_f. \quad (2.70)$$

Nous voyons que la tension de surface intervient comme une force singulière localisée sur l'interface (δ représente la distribution de Dirac). L'interface étant *a priori* dissociée du maillage fluide, la résolution de l'équation de quantité de mouvement (2.70) peut poser quelques problèmes. Pour remédier ces problèmes, nous régularisons la distribution de Dirac sur quelques mailles fluides. Pour ce faire nous régularisons la fonction d'Heaviside sur une épaisseur ϵ

$$H^\epsilon(\psi_f) = \begin{cases} 0 & \text{si } \psi_f < -\epsilon, \\ \frac{1}{2} \left(1 + \frac{\psi_f}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\psi_f}{\epsilon}\right) \right) & \text{si } |\psi_f| \leq \epsilon, \\ 1 & \text{si } \psi_f > \epsilon. \end{cases} \quad (2.71)$$

La distribution de Dirac régularisée est alors :

$$\delta^\epsilon(\psi_f) = \frac{dH^\epsilon(\psi_f)}{d\psi_f} = \begin{cases} 0 & \text{si } |\psi_f| > \epsilon, \\ \frac{1}{2\epsilon} \left(1 + \cos\left(\frac{\pi\psi_f}{\epsilon}\right) \right) & \text{si } |\psi_f| \leq \epsilon. \end{cases} \quad (2.72)$$

L'avantage de cette approche est sa grande simplicité de mise en œuvre. A partir d'une représentation level set de l'interface, il est aisé de simuler les changements topologiques. L'inconvénient majeur de cette approche est l'apparition de courants parasites au voisinage de l'interface. Ces courants se traduisent par des vitesses non physiques qui peuvent déstabiliser l'interface. De nombreuses études ont été apportées à cette problématique. Nous avons ajouté un terme de correction du gradient de pression pour obtenir un schéma dit *well balanced*.

Conditions de stabilité sur le pas de temps : le terme de tension de surface est traité comme une force supplémentaire dans les équations. Cette force engendre également une restriction sur le pas de temps qu'il faut ajouter aux conditions sur le pas de temps déjà obtenues (2.66). La condition complète est est :

$$\Delta t \left(C_c + C_\nu + \sqrt{(C_c + C_\nu)^2 + 4(C_g + C_s)} \right) \leq 2cfl, \quad (2.73)$$

avec $C_s = \frac{\sigma \|\kappa\|_\infty}{\min(\rho(\psi_f)) \min(\Delta x, \Delta y, \Delta z)^2}$ et $cfl \leq 1$. On prend généralement $cfl = 0,5$.

2.6.2 Interface raide et *Ghost Fluid Method* (GFM)

L'idée principale de la méthode GFM est de prolonger de façon continue de chaque côté de l'interface des variables discontinues en créant des cellules fantômes (Ghost cells) qui respectent les sauts à imposer (Kang *et al.*, 2000; Fedkiw *et al.*, 1999b; Liu *et al.*, 2000). Cette méthode repose alors sur l'utilisation de domaines fictifs. Nous avons choisi cette méthode par rapport à la méthode Immersed Interface car cette dernière ne préserve pas un opérateur symétrique comme c'est le cas pour la méthode Ghost Fluid. La méthode Ghost fluid a été introduite pour résoudre le problème elliptique suivant :

$$\nabla \cdot (\beta(\mathbf{x}) \nabla \psi(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.74a)$$

$$\psi(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega.$$

Le coefficient $\beta > 0$ est supposé discontinu à travers l'interface Γ . Les conditions de saut sont :

$$[\psi]_\Gamma = a(\mathbf{x}), \quad \mathbf{x} \in \partial\Gamma. \quad (2.74b)$$

$$[\beta\psi_n]_\Gamma = b(\mathbf{x}), \quad \mathbf{x} \in \partial\Gamma. \quad (2.74c)$$

avec $\psi_n = \frac{\partial\psi}{\partial n} = \nabla\psi \cdot \mathbf{n}$, \mathbf{n} étant la normale unitaire à l'interface. Sans rentrer dans les détails techniques qui peuvent être trouvés dans l'article de Kang *et al.* (2000), nous considérons que nous pouvons résoudre le système (2.74) de façon précise.

On rappelle en particulier que nous devons résoudre un problème elliptique similaire à (2.74a) lors de l'étape de correction de d'un algorithme de projection. Cette équation est :

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \psi \right) = -\frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \quad \text{avec } \psi = p^{n+\frac{1}{2}} - q. \quad (2.75a)$$

Le problème des écoulements bifluïdes avec discontinuité des masses volumiques et des viscosités réside dans le traitement du saut

$$[-pI + 2\mu D(\mathbf{u})] \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \text{ à travers } \Gamma_f.$$

Ce saut du tenseur des contraintes dans la direction normale peut être ré-écrit :

$$[p] - [2\mu(\nabla u \cdot \mathbf{n}, \nabla v \cdot \mathbf{n}, \nabla w \cdot \mathbf{n})\mathbf{n}] \cdot \mathbf{n} = \sigma \kappa \text{ à travers } \Gamma_f.$$

Le fluide étant incompressible nous avons $[(\nabla u \cdot \mathbf{n}, \nabla v \cdot \mathbf{n}, \nabla w \cdot \mathbf{n})\mathbf{n}] = 0$ et la condition de saut devient :

$$[p] = [2\mu](\nabla u \cdot \mathbf{n}, \nabla v \cdot \mathbf{n}, \nabla w \cdot \mathbf{n})\mathbf{n} + \sigma\kappa \text{ à travers } \Gamma_f. \quad (2.75b)$$

Il nous manque à ce stade uniquement une condition du type (2.74c). En considérant que la dérivée particulière du saut $[\mathbf{u}]$ est nulle, nous pouvons écrire :

$$\left[\frac{\nabla p}{\rho} \cdot \mathbf{n} \right] = \left[\frac{(\nabla \cdot (2\mu D(\mathbf{u}))^T \cdot \mathbf{n})}{\rho} \right] \text{ à travers } \Gamma_f. \quad (2.75c)$$

En utilisant $\psi = p^{n+\frac{1}{2}} - q$, nous pouvons réécrire les sauts pour la quantité ψ .

Le problème (2.75) est maintenant tout à fait similaire au problème elliptique modèle (2.74).

2.7 Interfaces fluide/fluide/structure et ligne triple

On appelle ligne triple¹⁵ de contact la ligne où se coupent les interfaces fluides/solide et fluide/fluide. Plusieurs méthodes ont été étudiées pour modéliser le déplacement de la ligne triple, notamment la méthode du film précurseur, la méthode d'interface diffuse et la méthode du modèle de glissement. Ces méthodes ont été largement décrites dans la thèse de Pinilla Velandia (2012).

Comme nous l'avons déjà vu précédemment, dans les méthodes d'interfaces diffuses, les quantités physiques peuvent varier très fortement à travers l'interface tout en restant continues. La ligne triple peut alors se déplacer grâce aux flux numériques. Le modèle le plus répandu est le modèle de Cahn-Hilliard

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \nabla \cdot (\gamma \nabla G), \quad (2.76)$$

où γ est un paramètre de mobilité et G un potentiel chimique. Cette équation est semblable à l'équation de transport de la fonction level set avec un terme diffusif supplémentaire. En général ϕ dénote une phase, à savoir $0 \leq \phi \leq 1$.

En ce qui concerne les méthodes de glissement, nous avons choisi la modèle macroscopique de Cox (1986). A l'équilibre, l'angle formé par l'interface fluide/fluide avec la surface solide est appelé angle de contact et est noté θ_E . En projetant l'équilibre des forces sur le plan solide, on obtient¹⁶ :

$$\cos \theta_E = \frac{\sigma_{s+} - \sigma_{s-}}{\sigma_{+-}}, \quad (2.77)$$

où σ sont les tensions de surface à l'interface des différents milieux. Si $\theta < \pi/2$ on parle de mouillage partiel par le fluide + et si $\theta > \pi/2$ le fluide est dit mouillant. Aux extrémités, si $\theta_E = 0$ on parle de mouillage total et si $\theta_E = \pi$ le fluide est non mouillant. L'angle θ_E est défini à l'équilibre lorsque l'interface est stationnaire. Si par contre l'interface se déforme, l'angle est dynamique et est noté θ_D . Si $\theta_D \neq \theta_E$, la ligne triple se déplace sous l'action de la force $F\theta_D = \sigma_{s+} - \sigma_{s-} - \sigma_{+-} \cos \theta_E$. Les détails de cette méthode sont bien expliqués dans la thèse de Pinilla Velandia (2012).

2.8 Le code de calcul NaSCar (NAvier-Stokes CARtésien)

2.8.1 Description

Le code de calcul que j'ai développé à l'INRIA se nomme NaSCar, en abréviation de Navier-Stokes Cartésien. Ce code est écrit en langage C/C++ et est couplé au standard MPI pour la parallélisation. Il utilise la librairie PETSC <http://www.mcs.anl.gov/petsc/> pour la résolution de systèmes linéaires en parallèle. Ce code permet de simuler différents types d'écoulements, en deux ou trois dimensions, tels que les interactions entre un fluide et des structures déformables (voire élastiques) ou encore des écoulements bifluïdes.

Le développement de ce code a pu être réalisé dans des conditions favorables grâce au cluster de calcul PLAFRIM (Plateforme Fédérative pour la Recherche en Informatique et Mathématiques), qui a été développé par une action INRIA PlaFRIM avec le soutien du LABRI et de l'IMB et d'autres partenaires : Conseil

15. Cette ligne se réduit à un point pour des écoulements en deux dimensions.

16. Nous faisons l'hypothèse ici que l'angle de contact est mesuré dans le fluide +.

Régional d'Aquitaine, FeDER, Université de Bordeaux et CNRS (voir <https://plafrim.bordeaux.inria.fr/>). Toutes les expérimentations numériques ont donc été effectuées sur PLAFRIM. La production de résultats a largement été effectuée sur le cluster de calcul du Mésocentre de Calcul Intensif Aquitain (MCIA, <http://www.mcia.univ-bordeaux.fr/>). Ces deux clusters possèdent à eux deux environ 4000 cœurs de calculs.

Le code est articulé autour des points suivants :

Librairies pour générer des profils d'obstacles

- Importation de géométrie discrète et maillage régulier de cette géométrie (pales d'éoliennes)
- Génération de profils 3D par B-splines (poissons, raies, ...)

Librairies interaction fluide/structure

- Calcul des forces et des moments
- One-way : déformation imposée des obstacles et déplacement par lois de Newton
- Two-way : couplage fortement implicite, structure poutre élastique

Librairies pour capturer et suivre les interfaces

- Génération d'une fonction level set (distance signée) à partir d'une géométrie quelconque
- Transport de la fonction level set (WENO5 et RK3 TVD)
- Redistanciation (méthode de type Godunov pour réaliser l'upwinding)
- Sélection de contour au besoin pour assurer la conservation du volume

Librairies pour la résolution des équations de Navier-Stokes

- Méthode de projection incrémentale de type prédicteur-correcteur ordre 2 (correction pression rotationnelle)
- Pénalisation
- Frontières immergées
- Méthode IPC hybride
- Bifluide avec tension surface : méthodes CSF et GFM
- Modèle de turbulence SGS Smagorinsky-Lilly et modèle de Vreman (2004)

Librairies d'exportation pour visualisation paraview

- Output généralisé : Choix des variables, pour snapshots, films, etc ...

Un dépôt APP est en cours.

Ce code est largement validé pour de nombreuses applications pour des nombres de Reynolds faibles et modérés, en deux et trois dimensions (Bergmann et Iollo, 2011; Bergmann *et al.*, 2014b). Excepté une application à affiner, les forts nombres de Reynolds n'ont pas réellement été étudiés, en partie du au fait que la méthode basée sur des grilles cartésienne ne s'y prête pas réellement. Ce problème reste ouvert (voir perspectives).

2.8.2 Validations

Nous avons effectué de nombreuses validations et calculs d'ordre (Bergmann et Iollo, 2011; Bergmann *et al.*, 2014b) pour des écoulements 2D. Nous avons bien obtenu l'ordre 2 en vitesse et pression (norme L_2) en espace pour la méthode IPC. L'ordre sur la pression tombe cependant à 1 pour la norme L_∞ . Dans ce qui suit nous allons simplement présenter une validation 3D. Beaucoup d'autres simulations et calcul de convergence seront effectués dans les chapitres suivants.

La sédimentation d'une sphère sous l'effet de la gravité a été largement étudiée dans la littérature, tant d'un point de vue expérimental (Clift *et al.*, 1978) que numérique (Glowinski *et al.*, 2001; Coquerelle et Cottet, 2008). La configuration de l'étude est donnée sur la figure 2.6. Nous choisissons les mêmes paramètres sans dimension que ceux introduits par Coquerelle et Cottet (2008), la sphère (de diamètre D) tombe dans un cylindre vertical (de diamètre $L = 1$) rempli d'un fluide de viscosité ν sous l'effet de la gravité $g = -980$. Nous modélisons un cylindre infini et imposons des conditions aux limites de Neumann homogènes ($\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \mathbf{0}$ où \mathbf{n} est le vecteur unitaire normal sortant) sur les frontières en haut et en bas du cylindre, respectivement Γ_T et Γ_B . Des conditions aux limites de non glissement, $\mathbf{u} = \mathbf{0}$, sont imposées par pénalisation (Angot *et al.*, 1999) sur la frontière latérale du cylindre Γ_L . La table 2.1 présente une comparaison entre les vitesses terminales pour différents diamètres et viscosités. La vitesse terminale obtenue dans notre étude, U , est comparée à des résultats expérimentaux U_E obtenus par Clift *et al.* (1978) et des résultats numériques U_G et U_C obtenus respectivement par Glowinski *et al.* (2001) et Coquerelle et Cottet (2008). Tous les résultats reportés sur la table 2.1 sont obtenus avec un pas de discrétisation en espace $h = 1/100$. Les résultats obtenus sont en très bons accords avec les résultats de la littérature.

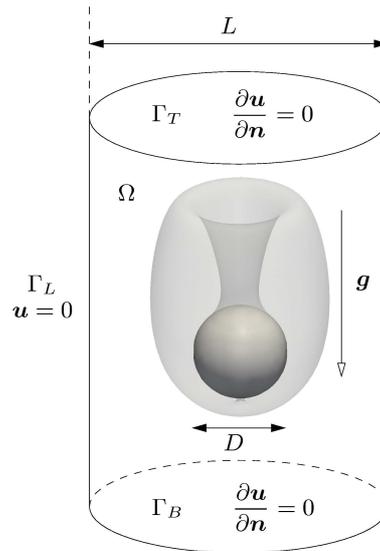


FIGURE 2.6 – Configuration de la sédimentation d’une sphère avec un exemple de champ de vorticit e.

D	ν	U_E	U_G	U_C	U
0.2	0.10	0.2571	0.2567	0.256	0.266
0.2	0.05	0.4603	0.4844	0.475	0.497
0.2	0.02	0.9129	0.9480	0.937	0.953
0.3	0.10	0.4047	0.4072	0.401	0.420
0.3	0.05	0.7493	0.7599	0.748	0.764
0.3	0.02	1.4359	1.3920	1.390	1.472

TABLE 2.1 – Comparaison entre les vitesses terminales de la s edimentation de sph ere pour plusieurs diam etres et viscosit es du fluide.

2.9 Conclusions

Ce chapitre avait pour but de pr esenter tous les outils que nous utiliserons pour  etudier les diverses applications qui seront pr esent ees dans les chapitres suivants. Tous ces outils ont  et e impl ement es dans le code de calcul massivement parall ele NaSCar que j’ai d evelopp e ces derni eres ann ees. Parmi ces outils figurent des m ethodes bien connues, et d’autres que nous avons sp ecifiquement d evelopp es pour certaines applications (IPC). Nous sommes encore en train de d evelopper d’autres m ethodes notamment pour  etudier des probl emes multi-physiques o u interviennent plusieurs effets  a diff erentes  echelles. En particulier, comme nous le d etaillerons par la suite, nous d eveloppons des mod eles d’interactions entre deux fluides et une structure  elastique (code sp ecifique pour la structure). Afin d’appr ehender les ph enomenes physiques qui interviennent  a diff erentes  echelles (couches limites, sillages) nous d eveloppons  egalement des raffinements de maillages de type quadtree/octree (grille cart esienne raffin ee en d ecoupant les cellules et en conservant la structure cart esienne), mais  egalement des m ethodes overset de type chim ere (une grille cart esienne plus une grille conforme  a l’obstacle). Cette partie qui concerne les maillages s’effectue en collaboration avec OPTIMAD engineering (politecnico di Torino).

Chapitre 3

Simulations d'écoulements autour d'obstacles déformables : application à la nage de poissons

Sommaire

3.1	Introduction	67
3.2	Préliminaires et définitions	68
3.2.1	Introduction	68
3.2.2	Configuration d'écoulement	68
3.2.3	Déformation et déplacement de corps	68
3.2.4	Classification de la nage	70
3.2.5	Représentation paramétrique de la nage de poissons	70
3.3	Simulations bidimensionnelles de nageurs	73
3.3.1	Choix de la résolution en espace	73
3.3.2	Mécanisme de la nage en 2D	74
3.3.3	Simulations numériques de nageurs de type BCF	74
3.3.4	Considérations énergétiques de la nage	77
3.4	Simulations tridimensionnelles de nageurs	79
3.4.1	Convergence de grille	80
3.4.2	Manœuvres : proie/prédateur	81
3.4.3	Efficacité en fonction des paramètres de nage	81
3.4.4	Efficacité en fonction de la rigidité de la nageoire caudale	82
3.4.5	Quelques autres nageurs	85
3.4.6	Simulations numériques de la nage d'un poisson à partir d'images	88
3.5	Conclusions et perspectives	95

3.1 Introduction

Afin de mettre en évidence l'efficacité des méthodes numériques mises en place de ce manuscrit, nous nous intéressons à la simulation numérique d'écoulement autour de corps déformables et en mouvement. Ce type de simulation serait en effet très difficile à réaliser en utilisant des outils classiques de simulation numérique basés sur des maillages adaptés à la géométrie des corps. Un remaillage serait nécessaire pour suivre les corps lors de leurs mouvements. La simulation autour de corps déformables sera mise en évidence sur le déplacement autopropulsé de corps assimilés à des nageurs. Différents types de nageurs seront étudiés, mais une très grande majorité sera de type poisson. Initialement, cette étude était limitée à illustrer le potentiel bénéfique de l'utilisation conjointe de maillages cartésiens, de méthodes de pénalisation et de fonctions level

set. Au fur et à mesure de l'évolution du travail et des simulations que nous avons effectuées, nous nous sommes intéressé de plus près à l'étude de la nage de poissons avec une attention particulière apportée à l'énergie dépensée.

3.2 Préliminaires et définitions

3.2.1 Introduction

La modélisation et la simulation numérique de la nage de poissons peut être d'un grand intérêt en biologie et dans les applications du domaine de l'ingénieur. En effet, mieux comprendre les mécanismes de la nage peut aider à clarifier certains aspects liés à l'évolution biologique et la physiologie de certains organismes aquatiques. Dans le domaine de l'ingénieur, l'étude et l'optimisation du transport sous marin peut aider à concevoir des véhicules sous marins performants ayant des capacités de manœuvre supérieures à celles actuellement développées. De façon plus générale, beaucoup d'autres applications allant de la sédimentation d'écoulement granulaire à la simulation d'un avion en pleine manœuvre peuvent bénéficier d'une modélisation précise des interactions instationnaires entre un fluide et des obstacles.

Les premiers résultats mathématiques concernant la nage de poisson sont ceux de Lighthill dans une série de papiers (Lighthill, 1960, 1969, 1970, 1971). Les principaux résultats sont obtenus dans une approche linéarisée pour des écoulements non visqueux. Depuis, plusieurs papiers ont suivi cette étude pour des fluides non visqueux, et une revue détaillée est donnée par Sparenberg (2002). Un exemple plus récent est présenté par Alben (2009), où une méthode numérique pour la dynamique d'un corps flexible dans un écoulement non visqueux est décrite.

Les premiers résultats prouvant le caractère bien posé du modèle d'interaction fluide-poissons ont été donnés par San Martin *et al.* (2008) et Nečasová *et al.* (2011).

En ce qui concerne la nage de poisson dans un écoulement visqueux il existe très peu de résultats numériques, la plupart des résultats connus sont obtenus expérimentalement. En particulier, Barrett *et al.* (1999) ont proposé un dispositif expérimental pour étudier la réduction de traînée dans la nage de poisson (le robot bluefin tuna). Un résultat marquant est que la puissance nécessaire pour nager est moindre que celle nécessaire pour tirer le robot immobile à la même vitesse. Un résultat encore plus marquant est qu'il est possible pour un poisson passif d'extraire de l'énergie d'une allée tourbillonnaire. En effet, Beal *et al.* (2006) ont montré qu'un poisson mort (anesthésié en fait) est capable de "nager" par couplage de son corps flexible avec les tourbillons.

Motivés par ces exemples, nous allons par la suite étudier la nage de poissons, sous différentes formes. Nous allons commencer par une étude en deux dimensions. Ensuite, nous étudierons brièvement le cas tridimensionnel, avec notamment d'autres nageurs de type méduses ou raies.

3.2.2 Configuration d'écoulement

Notre but est d'étudier l'écoulement incompressible autour de N_s corps déformables et autopropulsés. Le domaine considéré est noté $\Omega = \Omega_f \cup \Omega_i$, où le domaine Ω_f est rempli avec un fluide de densité constante ρ et de viscosité dynamique μ . Le domaine Ω_i représente le i -ème corps de densité ρ_i . La vitesse du corps $x_i \in \Omega_i$ est notée \mathbf{u}_i . La figure 3.1 présente une schématisation de la configuration d'écoulement étudiée.

En fonction des tests effectués, les équations de Navier-Stokes seront résolues soit sous forme dimensionnelle soit sous forme sans dimension en introduisant classiquement le nombre de Reynolds $Re = \frac{\rho U_{ref} D_{ref}}{\mu}$, où D_{ref} la longueur du poisson et U_{ref} une vitesse de référence. En particulier, nous prendrons un poisson de longueur $\ell = 1$, et choisissons alors $U_{ref} = 1$ et $D_{ref} = 1$. Le choix de $U_{ref} = 1$ se justifie *a posteriori* par le fait que les poissons étudiés nagent avec une vitesse approximativement égal à une longueur de corps par seconde. Notons enfin que la définition du nombre de Reynolds pour la nage de poisson n'est pas unique, et d'autres choix peuvent se présenter (Gazzola *et al.*, 2011).

3.2.3 Déformation et déplacement de corps

Nous sommes intéressés par des corps autopropulsés, *i.e.* les déplacements des corps ne sont pas imposés mais résultent de l'action des efforts exercés par le fluide sur les corps.

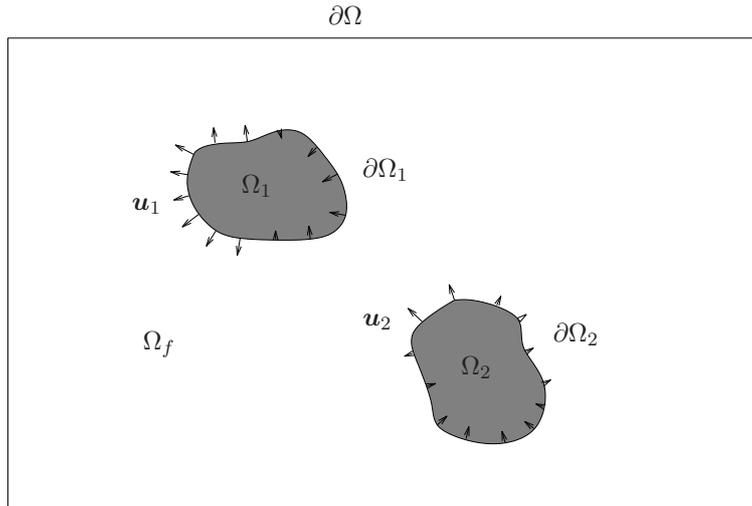


FIGURE 3.1 – Configuration d'écoulement générique.

Vitesses de translation et de rotation

La vitesse d'un corps immergé peut se décomposer comme étant la somme d'une vitesse de translation $\bar{\mathbf{u}}_i$ et d'une vitesse de rotation \mathbf{u}'_i relatives au centre de masse, plus une vitesse de déformation \mathbf{u}'_i . Dans la suite, la vitesse de déformation sera explicitement imposée, et les vitesses de translation et de rotation seront déterminées à l'aide des lois de Newton.

Soient \mathbf{F}_i et \mathcal{M}_i les forces et moments exercés par le fluide sur le i -ème obstacle. La vitesse de translation $\bar{\mathbf{u}}_i$ et la vitesse angulaire Ω_i s'écrivent :

$$m_i \frac{d\bar{\mathbf{u}}_i}{dt} = \mathbf{F}_i, \quad (3.1a)$$

$$\frac{dJ_i \Omega_i}{dt} = \mathcal{M}_i, \quad (3.1b)$$

avec m_i la masse du i -ème obstacle et J_i sa matrice d'inertie. A partir de la vitesse angulaire, la vitesse de rotation est $\mathbf{u}'_i = \Omega_i \wedge \mathbf{r}_i$ où $\mathbf{r}_i = \mathbf{x} - \mathbf{x}_i^G$ et \mathbf{x}_i^G le centre de masse.

Le tenseur des contraintes sans dimension est $\mathbb{T}(\mathbf{u}, p) = -p\mathbf{I} + \frac{1}{Re}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$. Soit \mathbf{n}_i le vecteur unitaire normal à $\partial\Omega_i$, les forces et moments sont :

$$\mathbf{F}_i = - \int_{\partial\Omega_i} \mathbb{T}(\mathbf{u}, p) \mathbf{n}_i d\mathbf{x}, \quad (3.2a)$$

$$\mathcal{M}_i = - \int_{\partial\Omega_i} \mathbf{r}_i \wedge \mathbb{T}(\mathbf{u}, p) \mathbf{n}_i d\mathbf{x}. \quad (3.2b)$$

On rappelle que la frontière $\partial\Omega_i$ ne coïncide *a priori* pas avec le maillage fluide. Puisque nous utilisons une méthode de discrétisation en espace d'ordre 2 (frontière immergée ou IPC), nous pouvons directement calculer ces quantités. En effet, une simple interpolation de Lagrange permet d'obtenir les valeurs de pression et de vitesse sur l'interface. Les vitesses étant d'ordre 2, les gradients seront d'ordre 1 d'où un calcul des forces et moments qui est consistant. Si par contre nous utilisons une méthode de pénalisation à l'ordre 1, les gradient à l'interface ne sont plus consistants, et une autre méthode de calcul est utilisée (voir Bergmann et Iollo (2011)).

La déformation des nageurs considérés ici est imposée. La force nécessaire pour imposer cette déformation n'est pas prise en compte pour le moment. Cette force est considérée comme une force intérieure et est supposée découplée de toutes les forces extérieures. Par conséquent, la déformation relative doit être de telle sorte que les moments linéaires et angulaires soient uniquement dus à l'action des forces extérieures. Autrement dit, la classe des déformations admissibles \mathbf{u}'_i doit satisfaire :

$$\int_{\Omega_i} \mathbf{u}'_i(\mathbf{x}, t) d\mathbf{x} = 0, \quad (3.3)$$

$$\int_{\Omega_i} \mathbf{x} \wedge \mathbf{u}'_i(\mathbf{x}, t) d\mathbf{x} = 0, \quad (3.4)$$

où la densité du corps est supposée constante.

3.2.4 Classification de la nage

La nage de poisson est généralement ordonnée en différentes catégories qui sont fonction de la forme et du mode de locomotion. Ces catégories sont décrites par Webb (1984) : La première est appelée BCF (Body and Caudal Fin), la seconde est BCF transitoire (accélération, manœuvres, *etc*) et la troisième est MPF (Median and Paired Fin). Dans la suite de ce chapitre, nous nous intéresserons aux modes BCF pour une nage périodique ainsi que pour les manœuvres. Le mode BCF représenté sur la figure 3.2.4 est classé en différentes sous catégories, *i.e.* anguilliforme, subcarangiforme, carangiforme et thunniforme (Lindsey, 1978; Sfatiotakis *et al.*, 1999).

Après avoir défini la forme du poisson §3.2.5, nous modéliserons la nage BCF périodique §3.2.5 et la nage BCF pour les manœuvres §3.2.5.

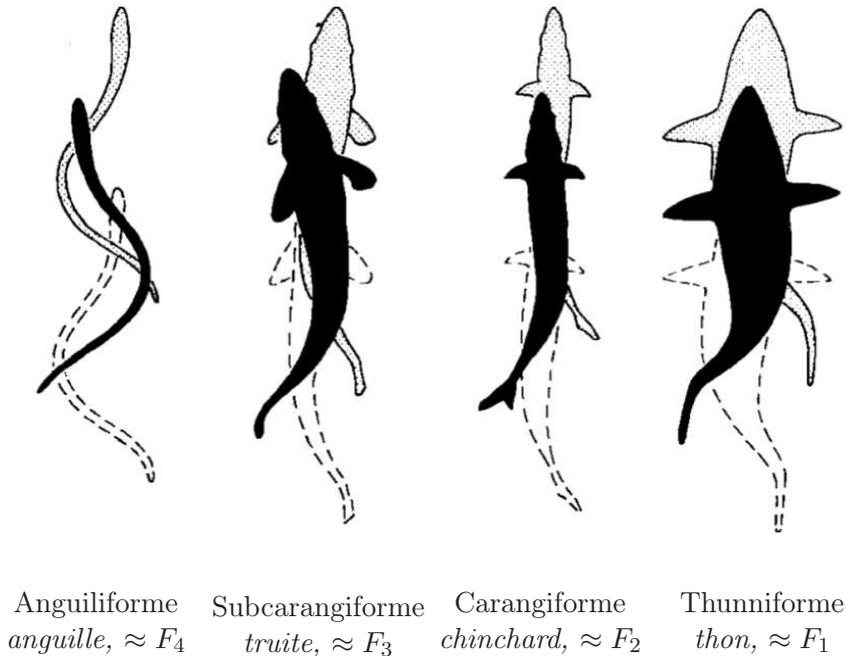


FIGURE 3.2 – Classification de la nage des poissons de type BCF (Body and Caudal Fin) par Lindsey (1978).

3.2.5 Représentation paramétrique de la nage de poissons

Modélisation de la forme du poisson

Nous commençons par définir une forme symétrique et stationnaire. Nous avons fait le choix d'approximer le profil 2D du poisson (vue de dessus) par une section d'une aile d'avion. Une méthode simple pour paramétriser une aile d'avion est d'utiliser une transformation conforme de Joukowski qui transforme un cercle de rayon $r_c = 1$ défini par un nombre complexe $\zeta = \eta + i\theta$ en un profil d'aile défini par le nombre complexe $z = x_s + iy_s$ dans le plan de transformation. La transformation est la suivante :

$$\xi = \zeta + \frac{1}{\zeta}. \quad (3.5)$$

Pour éviter que l'angle d'ouverture de la queue soit nul, nous avons opté pour une variante qui est la transformation de Karman-Trefftz (voir figure 3.3) :

$$z = n \frac{\left(1 + \frac{1}{\zeta}\right)^n + \left(1 - \frac{1}{\zeta}\right)^n}{\left(1 + \frac{1}{\zeta}\right)^n - \left(1 - \frac{1}{\zeta}\right)^n}, \quad (3.6)$$

où l'angle d'ouverture de la queue est $\alpha = (2 - n)\pi$. Une translation et contraction (ou extension) du plan z sont appliquées pour obtenir $0 \leq x \leq \ell$. A ce moment, la forme du poisson est entièrement définie par 4 paramètres, à savoir l'origine du cercle (η_c et θ_c), l'angle de la queue (α) et la longueur du poisson (ℓ). Puisque un profil symétrique est désiré ($\theta_c = 0$), nous choisissons $\eta_c < 0$, $\alpha > 0$ et $\ell \geq 0$. Différentes épaisseurs peuvent être obtenues en variant η_c . Par exemple, $\eta_c = -0.01$ approche un profil anguiforme, $\eta_c = -0.04$ approche un profil thunniforme, et $-0.05 < \eta_c < -0.01$ approche des profil carangiforme et subcarangiforme. Le profil pour

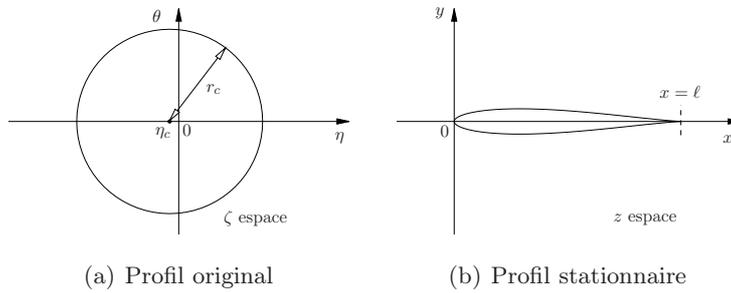


FIGURE 3.3 – Exemple de transformation de Karman-Trefftz.

un poisson 3D a été généré de façon différente. La colonne vertébrale du poisson est discrétisée en vertèbres x_i pour $i = 1, \dots, N$. Non déformée, la colonne vertébrale coïncide avec $0 \leq x \leq 1$, $y = z = 0$. Le profil 3D est composé avec N ellipses d'axes $y(x_i)$ et $z(x_i)$. Les axes $y(x)$ et $z(x)$ sont calculés avec des B-splines (voir figure 3.4). La coupe 2D du profil se rapproche toujours d'un profil d'aile d'avion. En particulier, nous utiliserons les dimensions maximum suivantes, $2y = 0.17$ et $2z = 0.24$. La nageoire caudale a une hauteur maximum de $2z = 0.25$. Ces paramètres ont été imposés pour s'approcher du profil du robot bluefin tuna du MIT (Barrett *et al.*, 1999). Un exemple de nage 3D est représentée sur la figure 3.6. Au besoin, d'autres profils seront utilisés.

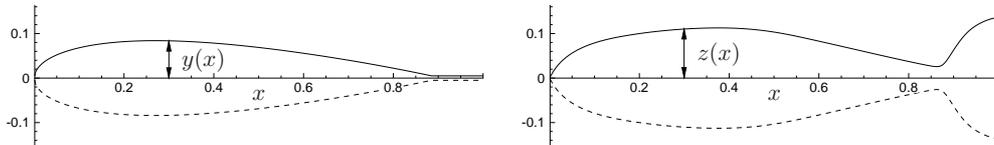


FIGURE 3.4 – Représentation des axes des ellipses axes $y(x)$ et $z(x)$ qui définissent le profil 3D.

Modélisation de lois de nage périodique de type BCF

Nous allons maintenant définir une loi de nage périodique. L'idée est de déformer la colonne vertébrale du profil stationnaire (défini par $0 \leq x \leq \ell$, $y = 0$ et $z = 0$) avec une loi $y(x, t)$. Il est nécessaire de conserver la contrainte $\ell = 1$. Soit s l'abscisse curvilinéaire de la colonne vertébrale déformée ($0 \leq s \leq \ell$). Nous fixons $s = 0$ à $x = x_0$. Pour une coordonnée s , l'abscisse $x(s)$ est déterminée en inversant la relation

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

La déformation correspondante $y(x(s), t)$ peut être évaluée numériquement. Comme décrit par Barrett *et al.* (1999), une loi de nage naturelle $y(x, t)$ est :

$$y(x, t) = a(x) \sin(2\pi(x/\lambda + ft)), \quad (3.7a)$$

où l'enveloppe est donnée par

$$a(x) = c_0 + c_1x + c_2x^2. \quad (3.7b)$$

En considérant que la longueur du poisson reste constante, $\ell = 1$, nous avons $x(\ell) \leq \ell$.

La loi est définie avec une vitesse de phase constante $c_p = \lambda f$, où λ et f sont la longueur d'onde et la fréquence des oscillations. La longueur d'onde λ n'est pas nécessairement égale à la longueur du poisson (Deng *et al.*, 2007). Cette loi de nage peut être centrée au nez du poisson (Deng *et al.*, 2007) ou à un certain pourcentage δ de la longueur du corps (Zhu *et al.*, 2002). Les paramètres c_0 , c_1 et c_2 peuvent alors être ajustés pour imposer une amplitude maximale de la nageoire caudale $A/2$ qui est un paramètre important pour l'efficacité de la nage (Lighthill, 1970). En pratique nous utilisons une transition durant une courte période entre le profil non déformé $y(x, t) = 0$ et le profil déformé avec $y(x, t) = a(x) \sin(2\pi(x/\lambda + ft))$. Une configuration de nageur 2D est donnée sur la figure 3.5(a) et une configuration 3D est donnée sur la figure 3.6

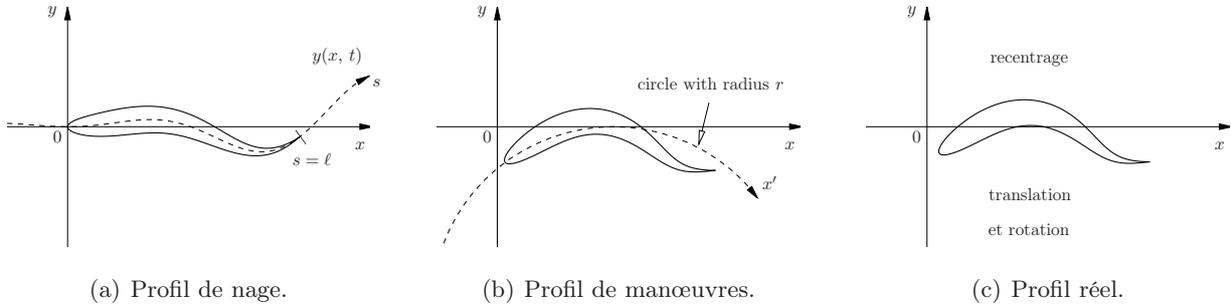


FIGURE 3.5 – Configuration des profils de nage et de manœuvres.

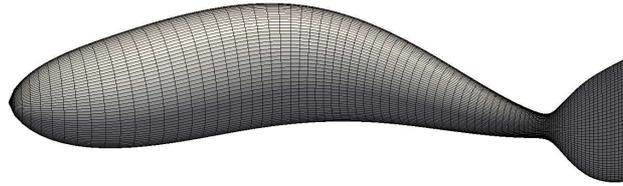


FIGURE 3.6 – Exemples de profils de nage.

Dans la suite nous notons \mathbf{b} les paramètres de forme du poisson (3 paramètres en 2D) et $\mathbf{s} = (c_0, c_1, c_2, \lambda, f, \delta)^T$ les paramètres de nage. Le profil instationnaire du poisson est noté $\mathbf{P}(\mathbf{b}, \mathbf{s}, t)$. Concrètement, nous procédons de la façon suivante :

1. imposer une déformation de la colonne vertébrale,
2. soustraire le déplacement du centre de masse pour vérifier (3.3),
3. effectuer une rotation opposée à celle induite par la déformation pour vérifier (3.4),
4. effectuer une homothétie si nécessaire (pour conserver la masse).

Modélisation de lois de nage de type BCF pour les manœuvres

Puisque nous avons défini une loi de nage ayant une courbure moyenne égale à zéro, les manœuvres peuvent sembler délicates à effectuer. Afin d'améliorer la manœuvrabilité, nous introduisons une stratégie qui consiste à modifier la loi de nage $y(x, t)$ pour obtenir une déformation moyenne non nulle. Nous choisissons

h	\bar{u}	d (%)	t_{CPU} (minutes)
1/32	-0.62	50	8
1/64	-0.81	14.8	42
1/128	-0.9	3.3	298
1/256	-0.93	—	2552

TABLE 3.1 – Evolution de la vitesse moyenne \bar{u} , de la déviation relative d et du temps CPU t_{CPU} en fonction du pas d'espace h .

la transformée $\mathcal{R} : P(\mathbf{b}, \mathbf{s}, t) \mapsto P_r(\mathbf{b}, \mathbf{s}, t)$ telle que le profil de nage $P(\mathbf{b}, \mathbf{s}, t)$ s'enroule sur une cercle de rayon r pour générer un nouveau profil $P_r(\mathbf{b}, \mathbf{s}, t)$ (voir figure 3.5(b)). Nous avons $P_\infty(\mathbf{b}, \mathbf{s}, t) = P(\mathbf{b}, \mathbf{s}, t)$. Le rayon minimal r peut varier en fonction du poisson considéré (Blake, 2004). Par exemple, $r = 0.47\ell$ pour le thon à queue jaune (Blake *et al.*, 1995), $r = 0.13\ell$ pour le dauphin *Corypaena hippurus* (Webb et Keyes, 1981) et même $r = 0.065\ell$ pour le poisson angelfish *Pterophyllum eimekei* Ahl (MPF) (Domenici et Blake, 1991). Comme pour la nage périodique, le profil final est recentré et réorienté pour éviter l'ajout de forces et moments artificiels. Un exemple du profil 2D $P_r(\mathbf{b}, \mathbf{s}, t)$ est donné sur la figure figure 3.5(c).

Remarques pour les simulations numériques

Les simulations numériques présentées dans ce chapitre ont été effectuées avec les méthodes décrites au chapitre précédent. Les paramètres de la simulation seront introduits au besoin dans chaque sous-section. Notons néanmoins que les simulations de poissons sont effectuées dans ce que l'on peut appeler aquarium, *i.e.* les conditions aux limites sont de type Dirichlet homogènes pour toutes les composantes de vitesses. On impose donc $\mathbf{u} = \mathbf{0}$ sur toutes les frontières extérieures du domaine de calcul. Les domaines de calcul sont des boîtes (2D ou 3D) de taille minimale, mais telle que la sensibilité des résultats en fonction de la taille du domaine soit négligeable.

3.3 Simulations bidimensionnelles de nageurs

Cette section est largement inspirée de Bergmann et Iollo (2011). Ce travail correspond au début de nos travaux autour de la modélisation et la simulation en mécanique des fluides. Nous avons utilisé une pénalisation classique à l'ordre 1.

Dans ce qui suit nous avons utilisé un pas de discrétisation $h = \Delta x = \Delta y$. Le pas de temps est choisi pour vérifier la condition CFL décrite au chapitre précédent. La résolution spatiale est basée sur l'étude de convergence suivante.

3.3.1 Choix de la résolution en espace

Le calcul d'écoulement autour d'obstacles rigides qui se déplacent a déjà été validé au chapitre précédent, ainsi que dans Bergmann et Iollo (2011) et Bergmann *et al.* (2014b). Nous sommes maintenant intéressés à l'étude de la convergence de grille pour des obstacles déformables. Cette étude se fera sur la vitesse limite qu'atteint le poisson. Cette quantité est intéressante car elle fait appel à plusieurs quantités sensibles comme le calcul des forces et des moments agissants sur le corps. Nous choisissons des paramètres de profil \mathbf{b} et une loi de nage \mathbf{s} (ce poisson est du type F_1 , voir la section suivante 3.3.3). Nous effectuons des calculs pour différentes valeurs de h pour $Re = 1000$. Ce poisson nage sur une distance égale à 9 et l'évolution de la vitesse u_h pour $0 \leq t \leq 10$ est présentée sur la figure 3.7. Le maillage le plus fin est obtenu pour $h = 1/256$. Pour le domaine considéré $[-10, 2] \times [2, 2]$ nous avons $3072 \times 1024 \simeq 3.1$ millions de nœuds. La solution \bar{u}_{256} est alors prise comme solution de référence. La déviation relative de la vitesse moyenne finale \bar{u}_h est alors définie par $d(h) = \|\bar{u}_h - \bar{u}_{256}\|_2 / \|\bar{u}_h\|_2$. la vitesse limite \bar{u}_h , la déviation relative $d(h)$ et le temps de calcul t_{CPU} (sur un processeur) sont reportés dans la table 3.1. La déviation relative est divisée par 4 quand h est divisé par 2. La table 3.1 montre que $h = 1/128$ est un bon compromis entre la déviation relative (seulement 3.3%) et le temps de calcul (8 fois moins que pour $h = 258$). Nous choisissons donc $h = 1/128$ pour la suite de nos simulations à $Re = 1000$.

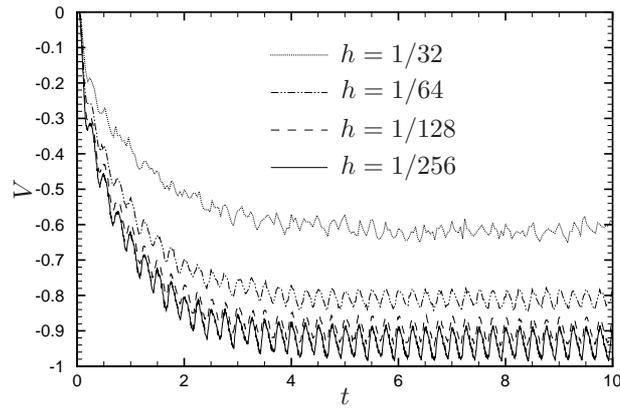


FIGURE 3.7 – Convergence de la vitesse de nage en fonction du pas d'espace h .

3.3.2 Mécanisme de la nage en 2D

Le mécanisme de la nage est représenté sur la figure 3.3.2. Une allée de Von Karman inversée est générée ce qui induit la formation d'un écoulement de type "jet" qui propulse le nageur.

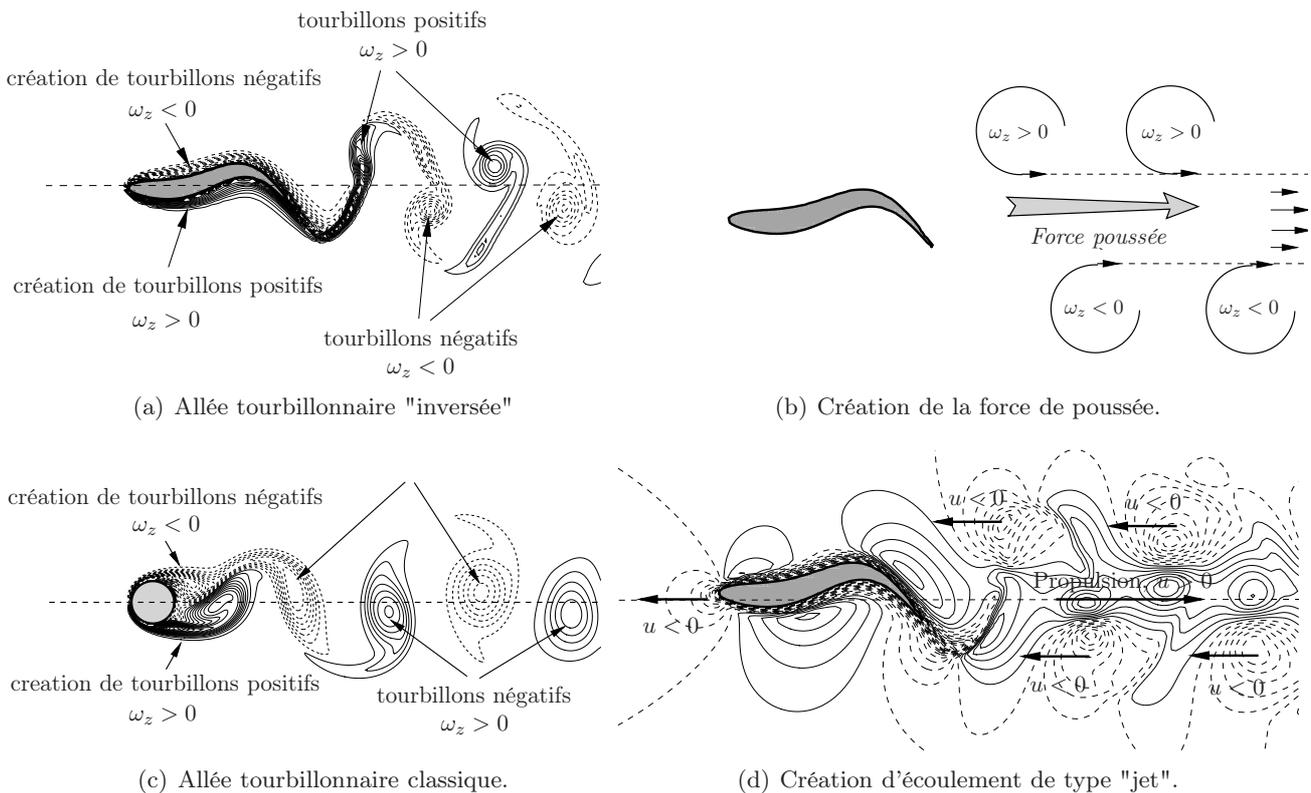


FIGURE 3.8 – Schématisation 2D des mécanismes de la nage d'un poisson.

3.3.3 Simulations numériques de nageurs de type BCF

Le but de cette section est de présenter une étude pour la nage de type BCF. Tout d'abord nous allons nous intéresser à la vitesse maximale atteinte sans considération de puissance dépensée. Comme nous l'avons déjà mentionné, il existe quatre types de modes BCF : anguilliforme, subcarangiforme, carangiforme et thunniforme.

Le mode anguilliforme présente de larges oscillations sur tout le corps effilé du nageur, et le mode thunniforme, à l’opposé, présente uniquement des oscillations localisées vers la queue d’un corps plus épais. Les deux autres modes présentent des caractéristiques intermédiaires. Ces modes sont modélisés avec la paramétrisation introduite à la section §3.2.5. Comme mentionné à la section précédente, il existe deux catégories de modes BCF : le mode périodique et le mode transitoire.

Mode BCF pour la nage périodique

La nage périodique présente un caractère instable dans le sens où tout le déplacement est influencé par le battement initial de la queue, générant un déplacement latéral non nul en moyenne. Un traitement spécifique doit alors être apporté comme allons le présenter dans la suite. Afin d’obtenir un déplacement latéral à moyenne nulle, nous corrigeons la trajectoire par ajout d’un rayon de courbure r qui est proportionnel à la déviation latérale du centre de mass Δy_G , *i.e.* $r \propto \Delta y_G$, avec $y_G(t = 0) = 0$.

Chaque mode de nage est modélisé avec différents profils et loi de nages. Comme décrit à la section §3.2.5, les modes sont définis par les paramètres \mathbf{b} et \mathbf{s} . On choisit quatre jeux de paramètres représentant quatre poissons F_1 , F_2 , F_3 et F_4 , qui modélisent les modes thunniforme, carangiforme, subcarangiforme et anguilliforme, respectivement. Ces paramètres sont synthétisés dans la table 3.2. Nous avons choisi d’utiliser la même angle de queue $\alpha = 5$, la même longueur $\ell = 1$, la même fréquence $f = 2$, $\delta = 0$ et $c_0 = 0$. Uniquement l’épaisseur du poisson défini par η , l’enveloppe de nage (définie par c_0 , c_1 et c_2) et la longueur d’onde λ sont modifiées. Les paramètres c_0 , c_1 et c_2 sont réajustés pour obtenir $A = 0,4$. Toutes les simulations sont

Poissons	Profil			loi de nage				
F_i	η_c	α	ℓ	c_1	c_2	λ	f	δ
F_1	-0.04	5	1	0.1	0.9	1.25	2	0
F_2	-0.03	5	1	0.4	0.6	1.00	2	0
F_3	-0.02	5	1	0.7	0.3	0.75	2	0
F_4	-0.01	5	1	1.0	0.0	0.50	2	0

TABLE 3.2 – Paramètres utilisés pour modéliser les quatre modes de nage.

indépendantes et sont stoppées quand le nageur a parcouru une distance égale à 9 fois la longueur de son corps. Le temps associé est noté T_9 . Les sillages pour $Re = 10^3$ sont représentés sur la figure 3.9. Des allées tourbillonnaires inversées de Von Karman sont visibles.

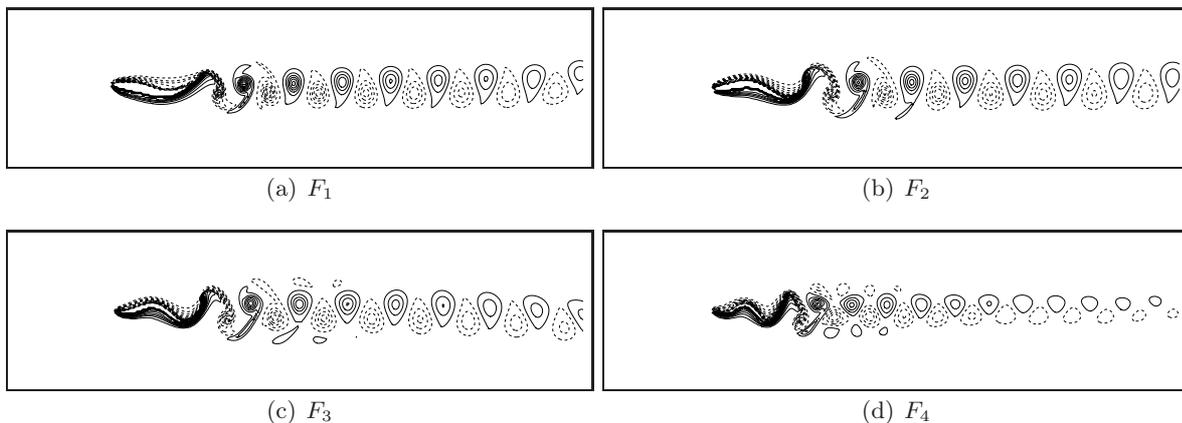


FIGURE 3.9 – Représentation du sillage généré par des nageurs de type BCF pour $Re = 10^3$. $-100 \leq \omega_z \leq 100$.

Le temps T_9 nécessaire pour parcourir la distance $d = 9$, la vitesse limite maximale U_{max} , l’accélération maximale γ_{max} et la vitesse moyenne \bar{U} pour $Re = 10^3$ et $Re = 10^4$ sont reportés dans la table 3.3 . L’accélération maximale est naturellement dépendante du régime transitoire imposé pour de $t = 0$ à $t = 0.2$.

	$Re = 10^3$				$Re = 10^4$			
<i>fish</i>	$ U_{max} $	$ \bar{U} $	$ \gamma_{max} $	T_9	$ U_{max} $	$ \bar{U} $	$ \gamma_{max} $	T_9
F_1	0.91	0.83	3.3	10.81	1.42	1.22	3.4	7.37
F_2	0.97	0.93	4.6	9.70	1.39	1.27	4.9	7.06
F_3	0.92	0.89	7.5	10.13	1.18	1.14	8.0	7.88
F_4	0.65	0.63	9.5	14.2	0.81	0.79	10.4	11.4

TABLE 3.3 – Comparaison de quelques caractéristiques de nage pour $Re = 10^3$ et $Re = 10^4$.

Mode BCF pour virages et manœuvres

L'objectif de cette section n'est pas de comprendre le mouvement complexe de vrais poissons, mais uniquement d'étudier le mouvement pour un prototype de robot. Nous allons montrer l'efficacité d'un simple contrôle basé sur la déformation introduite au §3.2.5 et l'angle de vision θ_f . Cet angle est arbitrairement défini comme étant l'angle orienté entre ligne passant par la tête et le centre de masse avec la ligne passant par la tête et l'objectif ciblé (de la nourriture par exemple), comme illustré sur la figure 3.10. Si $\theta_f = 0$ (l'objectif est droit devant), nous voulons retrouver $P_r(\mathbf{b}, \mathbf{s}, t) \rightarrow P(\mathbf{b}, \mathbf{s}, t)$, *i.e.* $r \rightarrow \infty$. En pratique nous n'imposons pas $r \rightarrow \infty$, mais nous passons du profil $P_r(\mathbf{b}, \mathbf{s}, t)$ au profil $P(\mathbf{b}, \mathbf{s}, t)$. Pour des raisons physiques, nous imposons $|r| \geq \bar{r}$. Nous avons choisi $\bar{r} = 0.5$ ce qui représente approximativement le rayon d'un virage pour un thon (Blake *et al.*, 1995). Nous supposons que cette limite peut être atteinte pour $|\theta_f| \geq \bar{\theta}_f$. Dans cette étude on fixe $\bar{\theta} = \pi/4$. Nous avons également imposé la relation $r(\theta_f)$ pour $|\theta_f| \leq \bar{\theta}_f$. Nous avons choisi $r(\theta_f) = \bar{r} \left(\frac{\bar{\theta}}{\theta_f}\right)^2$, $\forall 0 < |\theta_f| < \bar{\theta}_f$.

$$r(\theta_f) = \begin{cases} \infty & \text{si } \theta_f = 0, \\ \bar{r} & \text{si } \theta_f \geq \bar{\theta}_f, \\ -\bar{r} & \text{si } \theta_f \leq -\bar{\theta}_f, \\ \bar{r} \left(\frac{\bar{\theta}}{\theta_f}\right)^2 & \text{sinon.} \end{cases} \quad (3.8)$$

Les résultats sont illustrés sur la figure 3.11 pour le nageur F_1 à $Re = 10^3$. Les positions initiales de la tête

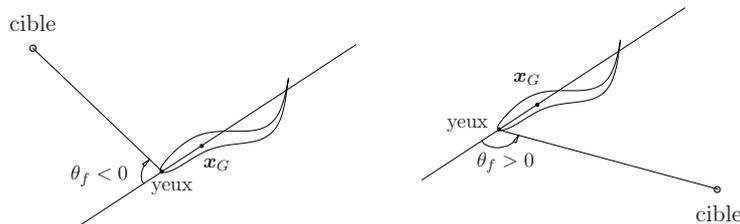


FIGURE 3.10 – Schématisation de l'angle de vision.

et de la queue sont $(0, 0)$ et $(\ell, 0)$ respectivement, et la nourriture est localisée en $(2, -1)$. L'angle initial de vision est alors $\theta_f = 2\pi - \arctan 1/2 \approx 0.85\pi$ tel que $r(t=0) = \bar{r}$. Nous imposons toujours une transition entre corps non déformé $r \rightarrow \infty$ et déformé $r = f(\theta_f)$. Nous venons de voir quelques mécanismes de nage de poisson sans considérations énergétiques de la nage. La prise en compte de la puissance dépensée au centre des sections suivantes.

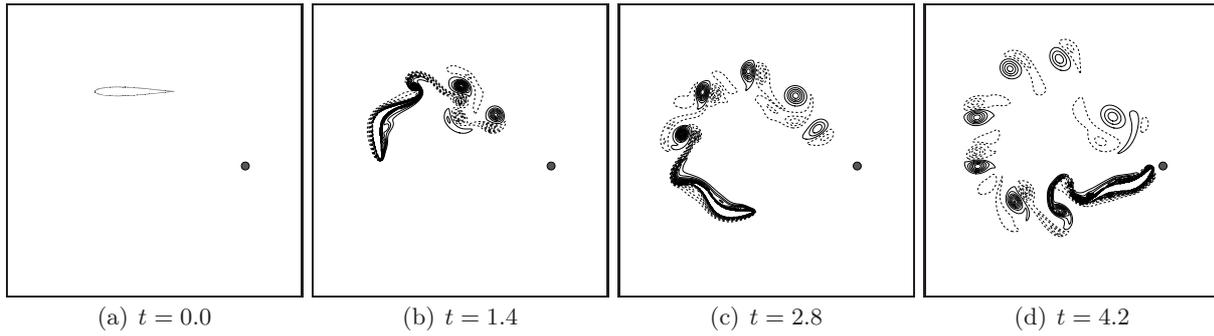


FIGURE 3.11 – Sillage et trajectoire du poisson qui manœuvre pour atteindre automatiquement la cible (disque gris) pour $Re = 10^3$. $-100 \leq \omega_z \leq 100$.

3.3.4 Considérations énergétiques de la nage

Puissance nécessaire pour nager

Un critère essentiel pour caractériser la nage est l'efficacité énergétique. On note $\sigma'_{ij} = \frac{1}{Re} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$. Sans dimension, la puissance nécessaire pour nager est :

$$P(t) = - \int_{\partial\Omega_i} p \mathbf{u} \cdot \mathbf{n} dS + \int_{\partial\Omega_i} (\sigma' \cdot \mathbf{n}) \cdot \mathbf{u} dS. \quad (3.9)$$

L'énergie nécessaire pour le poisson F_k pour atteindre la distance d est $E^{(k)} = \int_{T_k} P^{(k)} dt$, où T_k est le temps nécessaire pour atteindre la distance $d = 9$. Cependant, la quantité d'énergie dépensée par unité de temps dépend naturellement de la vitesse. Si tous les poissons avaient la même vitesse de nage U_r du poisson le plus lent (F_4), nous pourrions classer ces poissons par endurance. Pour ce faire, l'amplitude de la queue A est alors régulée à chaque itération : si $U > U_r$ nous diminuons A de façon proportionnelle, et vice versa. Nous notons F_1^r , F_2^r and F_3^r les nageurs F_1 , F_2 et F_3 avec le régulateur de vitesse. L'énergie nécessaire est maintenant reportée sur la table 3.4. Le poisson le plus efficace est F_1^r pour les deux nombres de Reynolds considérés.

<i>fish</i>	$Re = 10^3$	$Re = 10^4$
F_1^r	0.64	0.24
F_2^r	0.66	0.26
F_3^r	0.77	0.28
F_4^r	0.77	0.30

TABLE 3.4 – Comparaison de l'énergie $E^{(k)}$ dépensée pour nager sur une distance $d = 9$ pour $Re = 10^3$ et $Re = 10^4$. Les poissons F_1^r , F_2^r , F_3^r ont régulé l'amplitude de la queue pour nager à la vitesse du poisson F_4 .

La nage intermittente BAC (burst and coast)

Nous observons fréquemment le fait que les petits poissons ne nagent pas de manière continue, mais plutôt avec un caractère intermittent. Nous allons alors étudier si la nage intermittente, appelée *burst and coast* (BAC), permet d'obtenir des avantages énergétiques (Weihs, 1974; Videler et Weihs, 1982). La procédure utilisée est celle décrite par Weihs (1974). Nous supposons que à $t = t_0$ le corps rigide a une vitesse égale à U_i . Une période de nage commence de $t = t_0$ jusqu'à $t_1 = t_0 + T_b$ qui correspond au temps mis pour atteindre une vitesse maximale U_f . Une nouvelle période commence où la nage est arrêtée. Le poisson rigide glisse alors de la vitesse maximale U_f à t_1 à la vitesse minimale U_i à $t_2 = t_1 + T_c$. Une nouvelle période de nage de type BAC $T = T_b + T_c$ recommence. L'efficacité énergétique de la nage dépend *a priori* du choix de U_f et U_i (Weihs, 1974; Videler et Weihs, 1982). Dans ce qui suit nous étudions le nageur F_1 à $Re = 10^3$ et $Re = 10^4$.

A chaque nombre de Reynolds, quatre cas tests sont considérés. On note $U_f = \alpha_f U_{max}$ et $U_i = \alpha_i U_{max}$, où U_{max} est la vitesse maximale atteinte par F_1 avec la nage périodique pour $A/2 = 0.2$ (voir table 3.3). Nous considérons tout d'abord un grande différence entre U_f et U_i , *i.e.* nous choisissons $\alpha_f = 0.8$ et $\alpha_i = 0.2$. Ensuite, nous choisissons de faibles différences pour des fortes vitesses ($\alpha_f = 0.8$ and $\alpha_i = 0.6$), des vitesses moyennes ($\alpha_f = 0.6$ and $\alpha_i = 0.4$) et des faibles vitesses ($\alpha_f = 0.4$ and $\alpha_i = 0.2$). Pour la nage intermittente, nous considérons de rapides transitions linéaires entre phases de nage et phases de repos (et inversement). Nous notons R le ratio entre la puissance P_{bac} obtenue pour la nage intermittente (avec une vitesse moyenne \bar{U}) et la puissance pour la nage périodique P_{sps} (à la même vitesse \bar{U} en utilisant le régulateur de vitesse) :

$$R = \frac{P_{bac}}{P_{sps}}. \quad (3.10)$$

Dans chaque cas nous imposons également le correcteur sur la courbure moyenne de telle sorte que le poisson nage "droit". Les paramètres utilisés et les résultats sont reportés sur la table 3.5. Pour quasiment tous les

(α_i, α_f)	$Re = 10^3$	$Re = 10^4$
(0.2, 0.8)	0.77	0.85
(0.6, 0.8)	1.02	1.00
(0.4, 0.6)	0.85	0.81
(0.2, 0.4)	0.63	0.71

TABLE 3.5 – Efficacité R de la nage intermittente pour F_1 à $Re = 10^3$ et $Re = 10^4$ avec différents paramètres $U_f = \alpha_f U_{max}$ et $U_i = \alpha_i U_{max}$.

cas étudiés, la nage intermittente est plus efficace que la nage périodique. Cette efficacité augmente quand la vitesse de nage moyenne diminue. Ces résultats sont en très bon accord avec les résultats obtenus par Stöcker et Weihs (2001).

Influence de la nage en groupe sur l'efficacité énergétique

Il a été démontré que les poissons peuvent exploiter les tourbillons présents dans l'écoulement pour réduire l'énergie nécessaire pour se mouvoir (Liao *et al.*, 2003)). De façon plus surprenante, un poisson passif (anesthésié) est également capable de "nager" via une interaction avec les tourbillons (Beal *et al.*, 2006). Ce phénomène sera brièvement étudié et simulé au chapitre suivant. Le but de ce paragraphe est d'étudier les interactions entre plusieurs nageurs pour essayer de comprendre les mécanismes de nage en groupe qui amènent à une réduction de la puissance dépensée (Weihs et Webb, 1983). Tous les nageurs considérés dans la suite sont identiques (même profil et loi de nage). Soit D la distance latérale entre deux nageurs. Cette distance tend à diminuer pendant la nage à cause de l'effet venturi. Nous allons à nouveau utiliser un régulateur sur la courbure afin de garder cette distance fixe au cours du temps. On étudie tout d'abord deux nageurs initialement localisés aux points $(0, -D/2)$ et $(0, D/2)$. Nous considérons des lois de nage en phase et antiphase. Les figures 3.12(a) et 3.12(b) représentent les sillages associés en terme d'iso contours de vorticit   pour $D = 0.5$. Les isolateurs de vitesse horizontale u sont représentés sur les figures 3.12(c) et 3.12(d). Dans les deux cas il existe des zones de vitesses n  gatives (de m  me signe que la vitesse des poissons) dans le sillage g  n  r   par les deux poissons. Notons maintenant L la distance longitudinale entre ces poissons et un troisi  me plac   en $y = 0$ dans le sillage. L'amplitude de la queue du poisson plac   dans le sillage est r  gl  e pour nager    la m  me vitesse que les deux poissons de t  te. L'amplitude est diminu  e car ce poisson nage dans une eau entra  n  e par les deux premiers poissons. De ce fait, L et D restent approximativement constants pendant la nage. L'  nergie   conomis  e par le groupe de poissons pour $0.3 \leq D \leq 0.7$ et $2 \leq L \leq 3.5$    $Re = 10^3$ est report  e sur la table 3.6. La configuration la plus efficace est la nage en phase des deux premiers poissons. Pour la configuration en antiphase, les r  sultats d  pendent beaucoup de D et de L . Le troisi  me poisson doit nager en phase avec le sillage g  n  r   par les deux premiers poissons et pas n  cessairement en phase avec la loi de nage de ces m  mes poissons. Dans la limite $D \rightarrow \infty$ et/ou $L \rightarrow \infty$, il n'y a naturellement pas d'interactions entre les poissons et donc pas d'  nergie   conomis  e.

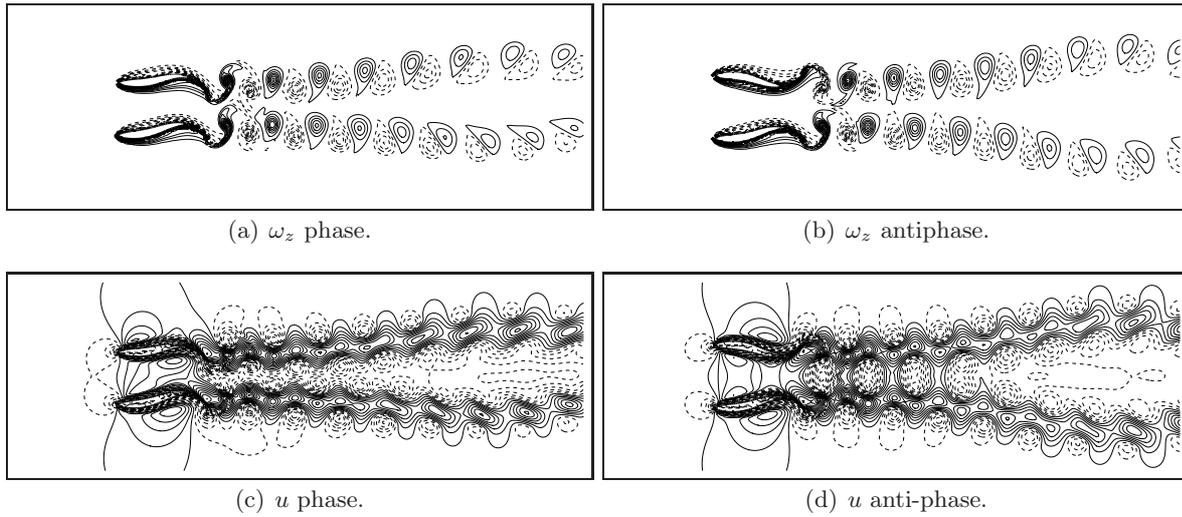


FIGURE 3.12 – Sillages générés par 2 poissons F_1 pour $Re = 10^3$ avec $D = 0.5$. Les figures de gauche et droite correspondent à la nage en phase et antiphase respectivement. $-100 \leq \omega_z \leq 100$, $-1.1 \leq u \leq 0.7$.

		Phase				Anti-phase			
L	D	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
1.5		15.0	16.3	11.1	7.1	6.8	6.9	9.8	7.1
2.0		10.1	14.5	9.8	6.0	6.8	6.1	9.8	6.0
2.5		8.4	13.6	9.0	5.1	6.7	5.3	9.0	5.1
3.0		15.0	15.1	6.9	5.0	5.2	5.1	7.0	3.2
3.5		5.2	13.2	6.2	2.2	4.9	5.0	6.2	0.5

TABLE 3.6 – Pourcentage d'énergie économisée par le groupe de trois poissons par comparaison avec trois nageurs indépendants à $Re = 10^3$.

3.4 Simulations tridimensionnelles de nageurs

Dans cette section nous allons nous intéresser plus particulièrement à des nageurs de type poisson thunniforme et leur efficacité énergétique, puis nous donnerons également quelques rapides exemples de nageurs de type méduse, raie, ou knifefish. Nous avons ici utilisé une méthode de frontière immergée à l'ordre 2 (IPC) et donc toutes les quantités intégrales comme les forces, les moments et la puissance dépensée ont pu être calculés directement par intégration sur le profil du poisson.

Le poisson considéré dans cette étude est celui représenté sur la figure 3.6 avec $\ell = 1$. Pour illustrer la nage d'un poisson 3D, nous avons utilisé la loi de nage (3.7) avec $\lambda = \ell$, $f = 2$, $c_0 = c_2 = 0$ et $c_1 = 0.1$. La figure 3.13 présente un exemple de sillage 3D généré par un poisson à $Re = 10^4$. Nous voyons que le sillage 3D est très différent du sillage 2D. En effet, la figure 3.14 qui représente une coupe 2D du sillage 3D montre que le sillage à une forme de "V" tandis que pour la nage 2D le sillage est plutôt rectiligne.

Dans cette partie consacrée à la nage 3D nous allons nous intéresser plus particulièrement à l'efficacité de la nage. L'efficacité propulsive peut être définie par (von Loebbecke *et al.*, 2009)

$$\eta = \frac{W_{utile}}{W_{total}}, \quad (3.11)$$

où W_{total} et W_{utile} sont le travail total et le travail utile généré par le nageur durant une période de nage (un battement de queue).

La surface du poisson $\partial\Omega_s$ est discrétisée de telle sorte que $\cup_{i=1}^N \partial\Omega_s^i = \partial\Omega_s$.

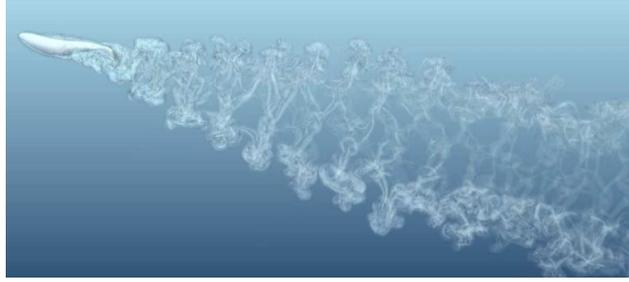


FIGURE 3.13 – Visualisation du sillage 3D (iso vorticité) généré par un poisson à $Re = 10^4$.

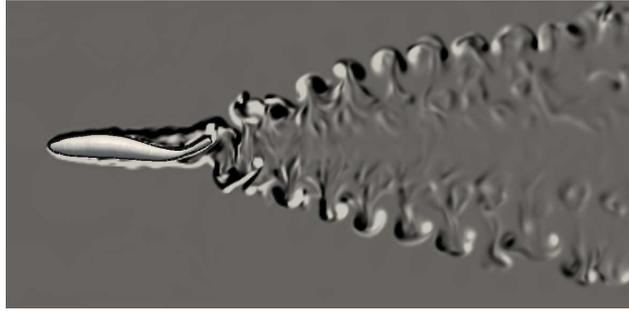


FIGURE 3.14 – Coupe 2D du sillage généré par une nage 3D.

La force exercée par le fluide sur la section surface $\partial\Omega_s^i$ est

$$\mathbf{F}^i = (F_x^i, F_y^i, F_z^i)^T = - \int_{\partial\Omega_s^i} \mathbb{T}(\mathbf{u}, p) \cdot \mathbf{n} \, d\mathbf{x},$$

où \mathbf{n} est la normale extérieure à $\partial\Omega_s^i$. Afin que les relations suivantes soient consistantes, le champ de pression p à l'infini est zéro. En d'autres termes, le champ de pression doit être considéré comme une variation de pression par rapport à l'infini (dépression ou surpression).

Le travail total fourni par le nageur pendant une période T est alors :

$$W_{total} = \int_T \sum_{i=1}^{N_s} \mathbf{F}^i \cdot \mathbf{u}^i \, dt, \quad (3.12)$$

où \mathbf{u}^i est la vitesse moyenne sur la surface $\partial\Omega_s^i$.

Le travail utile est défini comme étant la partie du travail total qui correspond aux forces exercées dans la direction de nage (von Loebbecke *et al.*, 2009). Par exemple, si la vitesse de nage dans la direction x est positive, $U_x > 0$, la force utile est $(F_x + |F_x|)/2$.

Le travail utile est alors :

$$W_{utile} = \int_T \sum_{i=1}^{N_s} \frac{F_x^i + |F_x^i|}{2} U_x^i \, dt. \quad (3.13)$$

Notons enfin que cette définition de l'efficacité d'un nageur n'est pas universelle et fait encore débat. Une autre définition est donnée par Borazjani et Sotiropoulos (2008).

3.4.1 Convergence de grille

Tout d'abord nous allons étudier l'influence du maillage sur la solution numérique. Nous allons effectuer une étude de convergence de maillage pour $Re = 1000$. Nous nous intéressons à deux quantités, à savoir la vitesse de nage $\langle v \rangle$ et l'efficacité $\langle \eta \rangle$ ($\langle \cdot \rangle$ est un opérateur de moyenne agissant sur deux périodes quand le régime asymptotique est atteint). L'évolution de la vitesse et de l'efficacité est représentée sur la figure 3.15

en fonction du pas d'espace $1/h$ (ici nous avons $h = \Delta x = \Delta y = \Delta z$). Ces deux quantités semblent converger vers des valeurs limites pour les maillages les plus fins. Cependant, le cas $h = 1/120$ est un bon compromis entre précision et coûts de calcul. En effet, il n'y a que 2% de différence entre $h = 1/120$ et $h = 1/160$, mais $h = 1/120$ nécessite la moitié du temps de calcul de $h = 1/160$. Dans les simulations qui vont suivre, nous prendrons donc $h = 1/120$.

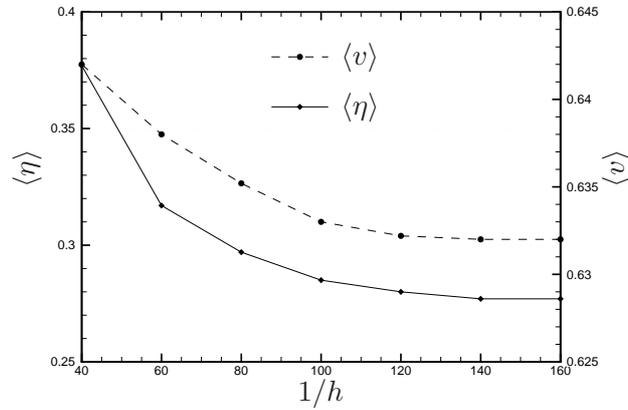


FIGURE 3.15 – Evolution de la vitesse moyenne $\langle v \rangle$ et de l'efficacité moyenne $\langle \eta \rangle$ en fonction $1/h$ (avec h le pas d'espace). Les crochets représentent un opérateur de moyenne.

3.4.2 Manœuvres : proie/prédateur

L'algorithme basé sur l'angle de vision défini sur la figure 3.10 est utilisé pour imposer la courbure. L'angle de vision est mis à jour à chaque instant en prenant compte de la position des deux poissons (proie et prédateur). La figure 3.16 présente un exemple de simulation numérique. Dans cet exemple, la proie nage de façon rectiligne et n'essaie pas d'échapper au prédateur. Il serait intéressant d'imposer une loi de nage pour échapper au prédateur (manœuvre brusque si la distance entre les deux poisson est inférieur à un seuil), et voir si, et quand, le prédateur peut atteindre sa proie.

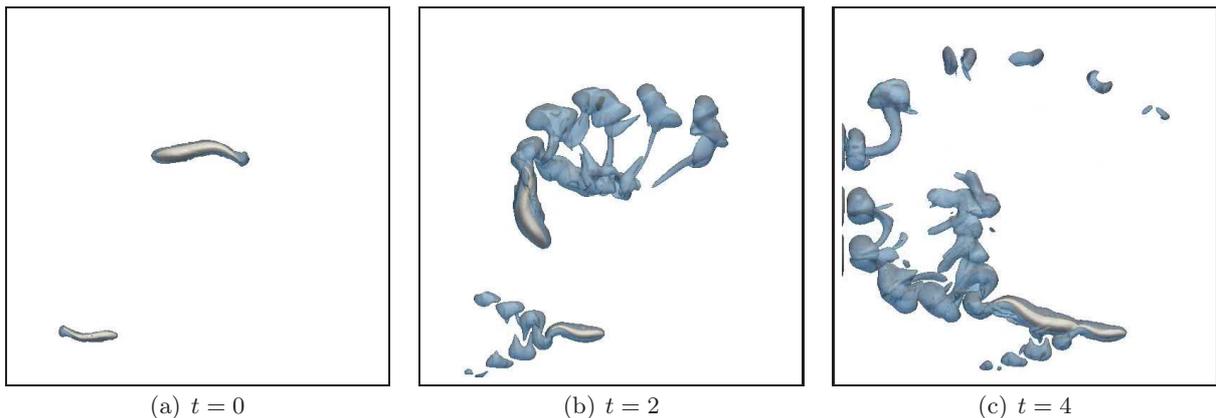
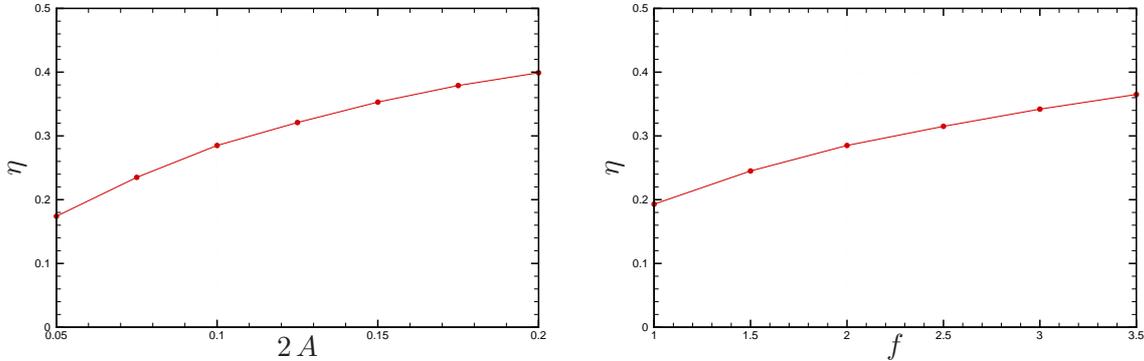


FIGURE 3.16 – Exemple de la chasse par un prédateur. La proie nage mais n'esquive pas le prédateur.

3.4.3 Efficacité en fonction des paramètres de nage

Nous étudions tout d'abord l'influence de la loi de nage (3.7) sur l'efficacité η . Nous considérons uniquement deux paramètres ouverts, la fréquence de nage f et l'excursion maximale de la queue A (pour $\ell = 1$ nous avons $A = 2c_1$). Nous commençons par fixer l'amplitude $A = 0, 1$. Plusieurs simulations sont alors réalisées

pour différentes fréquences comprises entre $f = 0,5$ et $f = 3,5$. Ensuite, nous fixons la fréquence à $f = 2$ et effectuons des simulations numériques pour différentes amplitudes allant de $A = 0,05$ à $A = 0,2$. Les résultats sont reportés sur les figures 3.17(a) et 3.17(b). L'efficacité maximale est obtenue pour les plus grandes valeurs



(a) Efficacité de nage η en fonction de l'amplitude A pour $f = 0.5$ (b) Efficacité de nage η en fonction de la fréquence f pour $A = 0.2$

FIGURE 3.17 – Evolution de l'efficacité de nage en fonction de l'amplitude et de la fréquence de battement de la queue.

de l'amplitude A et de la fréquence f . Les efficacités les plus grandes sont obtenues pour les vitesses de nage les plus grandes (pour le nombre de Reynolds considéré, $Re = 1000$). Cela peut paraître surprenant, mais ce phénomène est observé dans la nature : les petits poissons (assimilés à $Re = 1000$) ont des fréquences et amplitudes de nages importantes. Il peut alors être intéressant d'étudier une autre alternative pour améliorer l'efficacité pour une même loi de nage. Nous choisissons $A = 0.1$ et $f = 2$. La principale différence entre nos simulations et l'observation physique que l'on peut faire d'un vrai poisson réside dans le fait que la queue d'un vrai poisson est élastique. Nous verrons au chapitre suivant qu'un couplage élastique *two way* est très coûteux numériquement. Nous proposons alors un modèle qui permet de changer la rigidité de la queue de telle sorte qu'un couplage faible alterné est suffisant pour obtenir une bonne stabilité numérique.

3.4.4 Efficacité en fonction de la rigidité de la nageoire caudale

Modèle de rigidité de la nageoire caudale

L'objectif est d'étudier l'influence de la rigidité de la queue sur l'efficacité de la nage. La déformation peut être imposée sur toute la colonne vertébrale (voir figure 3.18), ou seulement sur une partie qui exclue la queue (environ 15 – 20% du corps). Dans ce dernier cas, nous modélisons la queue par un système discret masses/ressorts/amortisseurs (traits pointillés rouges sur la figure 3.18). La queue est alors composée par des segments rigides reliés entre eux par des joints élastiques. Chaque joint est soumis à un couple C_i que nous choisissons proportionnel au carré de la vitesse tangentielle locale V_{i+1} du joint suivant¹. Pour simplifier le modèle, on considère que les forces d'inertie de l'élément suivant sont négligeables. Le modèle est :

$$\ddot{\theta}_i + \beta\dot{\theta}_i + k\theta_i = C_i, \quad i = 1, \dots, N_L - 1$$

où θ_i est l'angle de rotation en fonction de l'élément précédent, $C_i = -\alpha|V_{i+1}|V_{i+1}$, N_L est le nombre de liaisons. Le point $i = 0$ correspond à $s = 0.8\ell$ (loi de nage (3.7)). La condition initiale est $\theta_i = 0, \forall i$. De ce fait, pour une queue extrêmement rigide nous avons $\theta_i = 0, \forall i$. Dans ce qui suit nous prenons $\beta = 1, k = 4 \cdot 10^3, 0 \leq \alpha \leq 16 \cdot 10^{-2}$ et $N_L = 60$. Ces paramètres sont choisis empiriquement pour obtenir des déformations que nous avons jugé réalistes, comme sur la figure 3.18.

Résultats numériques

Des expériences numériques montrent que des valeurs α comprises entre 0.02 (très rigide) et 0.16 (très mou) permettent d'obtenir des déformations réalistes. La variation de l'efficacité en fonction de α pour $f = 2$

1. Ce choix pour être discutable, mais est supporté par des expériences numériques.

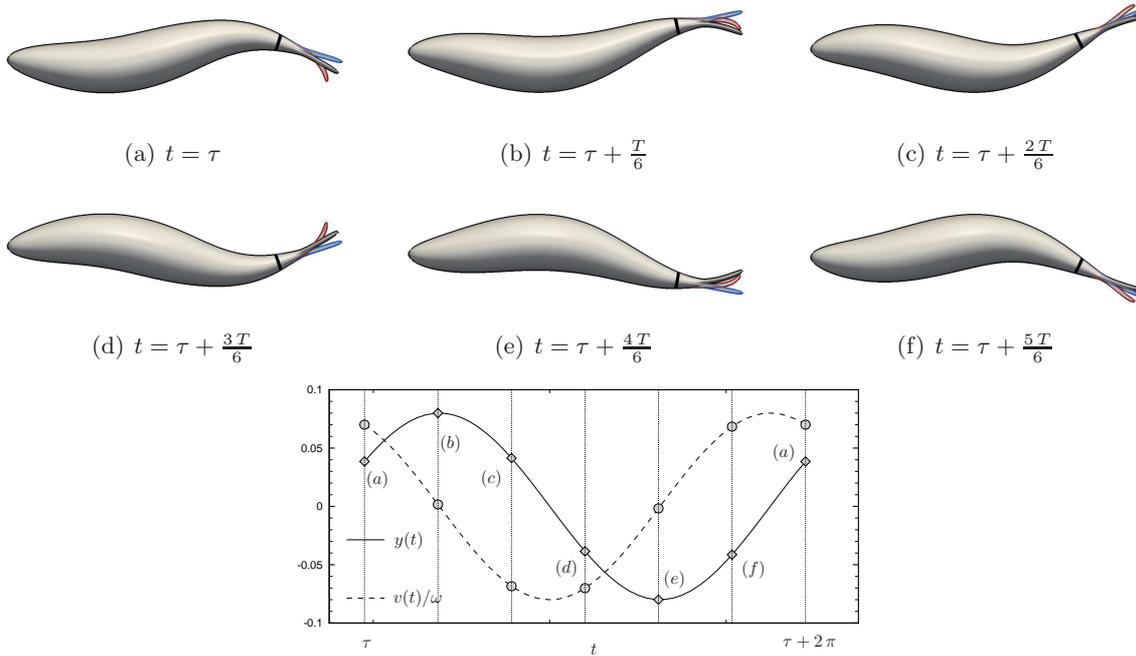


FIGURE 3.18 – Déformation du poisson sur une période T (fréquence $f = 2.0 \text{ Hz}$). La même loi de nage sinusoïdale est appliquée sur tout le corps sauf sur la nageoire caudale entre la section noire et la fin de la queue où un comportement élastique est simulé. Le corps noir correspond à une déformation imposée, le corps bleu correspond à une nageoire rigide et le corps rouge correspond à une nageoire flexible.

est représenté sur la figure 3.19(a). Une valeur $\alpha = 0.1$ maximise l'efficacité. Cette flexibilité de la queue permet d'augmenter l'efficacité d'environ 25% en comparaison avec le cas où la déformation (3.7) est imposée sur tout le corps. Le sillage généré avec $\alpha = 0.1$ est représenté sur la figure 3.30. Le sillage est composé de deux allées en "V" comme c'est le cas du sillage généré par une plaque oscillante (Li et Lu, 2012). Ceci est caractéristique pour des nombres de Strouhal de nage $St = \frac{fA}{V}$ élevés (Dai *et al.*, 2012), typiquement pour $St \geq 0.3$ (Borazjani et Sotiropoulos, 2009). Dans notre étude, le nombre de Strouhal minimal obtenu est de l'ordre de 0.5. Des nombres de Strouhal plus faibles (environ 0.2) peuvent être obtenus pour d'autres lois de nage avec des nombres de Reynolds plus élevés. Dans ces premières comparaisons, la vitesse du poisson est également influencée par la variation de α . Nous allons alors étudier la variation de l'efficacité pour des mêmes vitesses de nage. La fréquence de nage est toujours $f = 2.0$, mais l'amplitude de la queue est régulée comme dans le cas 2D. La vitesse cible est la vitesse la plus faible obtenue, *i.e.* $V = 0.4$ pour $\alpha = 0.16$. Finalement, l'excursion de la queue est à peu près constante pour tous les α , et les nombres de Strouhal de nage St sont presque constants. La figure 3.20(a) représente l'évolution de l'efficacité en fonction de α . Des paramètres $\alpha \geq 0.04$ sont efficaces par rapport aux cas rigides. Pour $\alpha = 0.12$ l'efficacité est augmentée de 35%. Cette nette augmentation pour $\alpha = 0.12$ s'explique par le fait que le travail utile augmente et le travail total diminue pendant une période (figure 3.20(b)). La figure 3.22 présente la distribution du travail utile sur le corps du poisson, dans le cas rigide ($\alpha = 2 \cdot 10^{-2}$) et dans le cas optimal ($\alpha = 1.2 \cdot 10^{-1}$). Dans le cas optimal, de plus grandes zones efficaces sont générées sur la queue.

Influence du nombre d'éléments de la nageoire caudale

Dans la section précédente, nous avons utilisé $N_L = 60$ éléments sur la queue. Dans ce qui suit, nous prenons $1 \leq N_L \leq 5$ (voir figure 3.23). Le cas $N_L = 60$ correspond à $\alpha = 0.12$. Le paramètre α est modifié pour $1 \leq N_L \leq 5$ afin que les portraits de phases de l'extrémité de la queue soient le plus proches possible du cas $N_L = 60$ avec $\alpha = 0.12$. On peut voir que le profil pour $N_L = 5$ est très proche de celui obtenu pour $N_L = 60$.

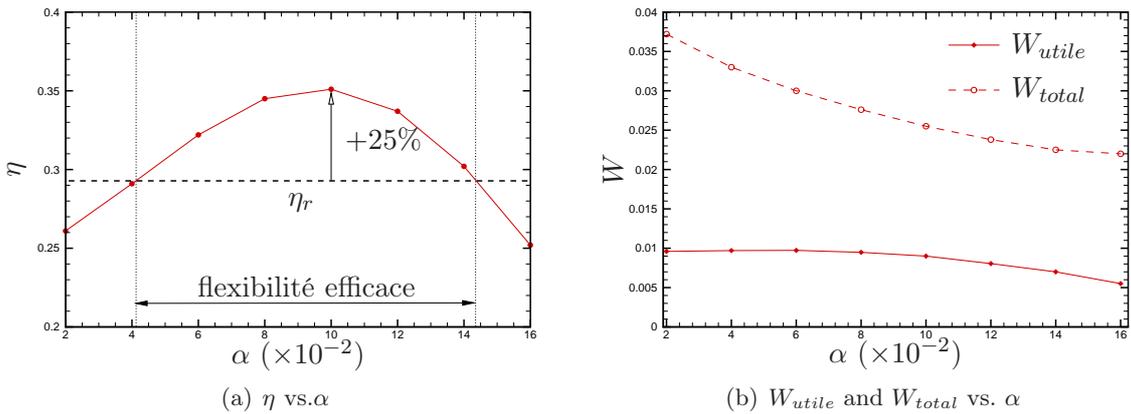


FIGURE 3.19 – Evolution de l'efficacité η , du travail utile W_{utile} et du travail total W_{total} en fonction du paramètre de flexibilité α . Les traits pointillés correspondent à une déformation imposée.

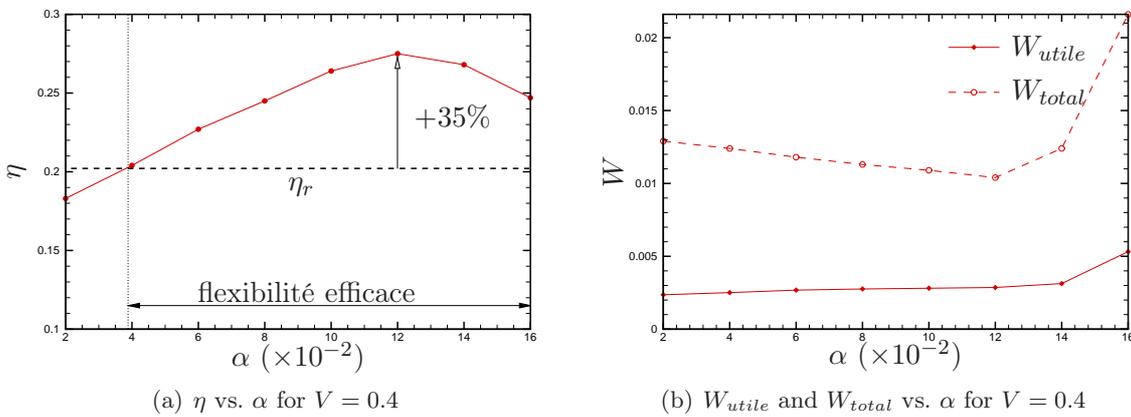


FIGURE 3.20 – Evolution de l'efficacité η , du travail utile W_{utile} et du travail total W_{total} en fonction du paramètre de flexibilité α pour $V = 0.4$. Les traits pointillés correspondent à une déformation imposée pour $V = 0.4$.



FIGURE 3.21 – Sillage généré par le poisson avec $\alpha = 0.1$ pour $Re = 10^3$.

La figure 3.24(a) montre que l'efficacité est améliorée quand N_L augmente. L'évolution du travail utile et du travail total en fonction de N_L est représenté sur la figure 3.24(b). Alors que la valeur du travail utile est presque constante avec N_L , la valeur du travail total diminue quand N_L augmente pour atteindre la valeur obtenue pour $N_L = 60$. Des paramètres optimaux pour la rigidité de la queue permettent de diminuer le travail fourni par le nageur, spécifiquement dans la direction latérale. En effet, quand la vitesse de nage est constante, le travail total est égale au travail fourni pour les mouvement latéraux (le travail utile et son dual "inutile" s'annulent réciproquement).

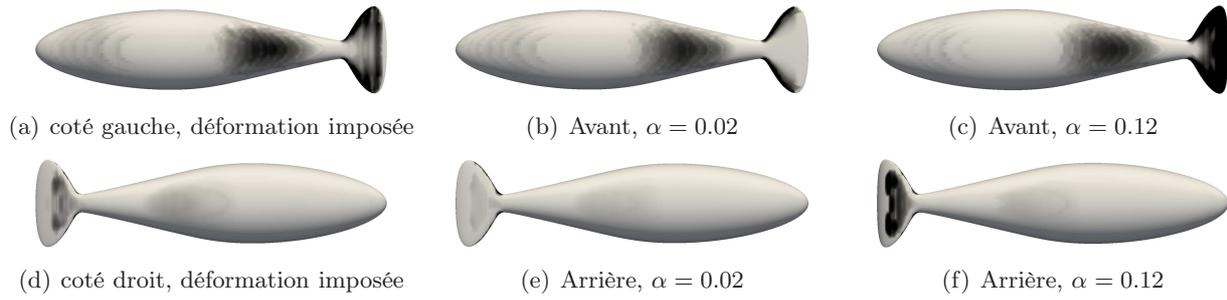


FIGURE 3.22 – Visualisation de la contribution du travail utile sur le corps du nageur pour une déformation imposée (gauche), pour une nageoire rigide ($\alpha = 0.02$, milieu), et pour une nageoire flexible ($\alpha = 0.12$, droite) pour $V = 0.4$. Les clichés sont extraits aux mêmes instants. Couleurs : gris (0, zones inefficaces), noir (-0.05 , zones efficaces).

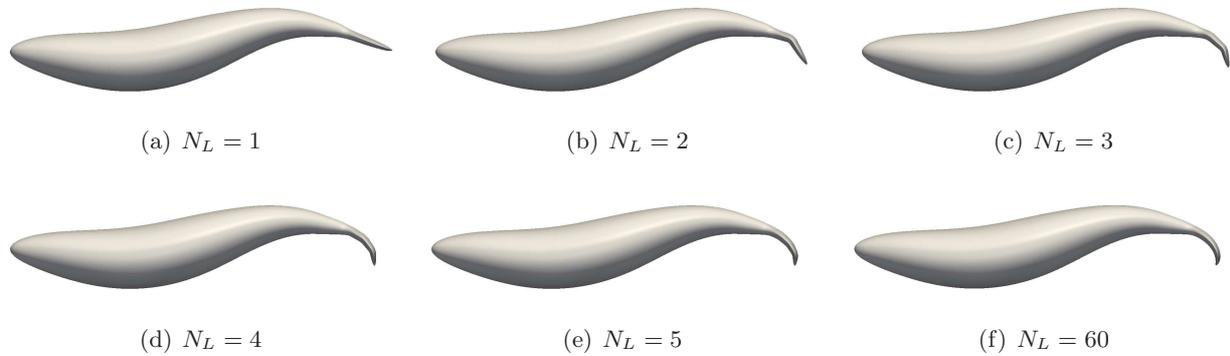


FIGURE 3.23 – Profil de poisson pour différents nombres de liaisons de la nageoire caudale N_L . Le cas $N_L = 60$ correspond à $\alpha = 0.12$.

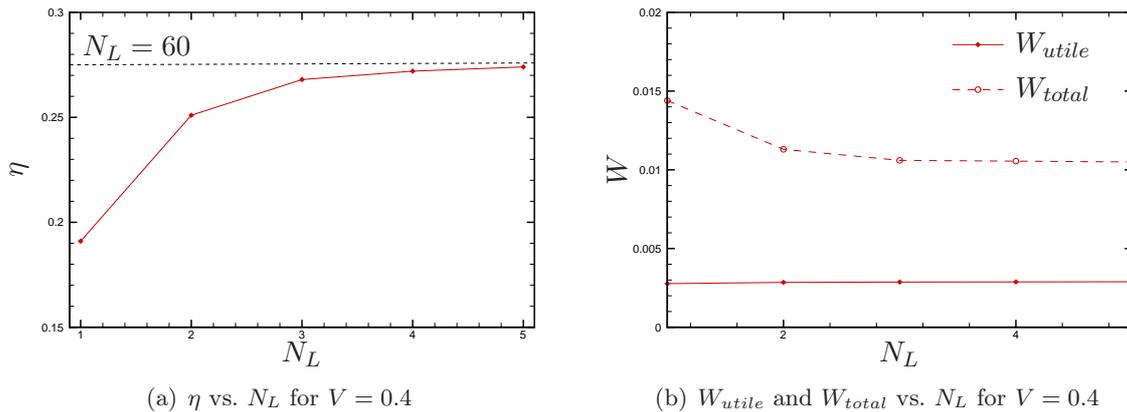


FIGURE 3.24 – Evolution de l'efficacité η , du travail utile W_{utile} et du travail total W_{total} en fonction du nombre de liaisons N_L pour $V = 0.4$. Les trait en pointillés correspond à la déformation imposée sur tout le profil pour $V = 0.4$

3.4.5 Quelques autres nageurs

Ce paragraphe a pour but de présenter brièvement quelques autres nageurs, avec une attention particulière portée à un prototype de méduse. Nous verrons également la nage d'une raie et d'un knifefish.

Méduse

Nous étudions ici l'écoulement généré par un prototype de méduse composé d'une lentille de diamètre $D = 4\text{ cm}$ et d'épaisseur maximale $\ell = D/10$. Le profil de nage est obtenu en déformant la lentille. Cette lentille est alors enroulée sur une sphère de diamètre $r(t)$. Dans cette étude $r(t)$ oscille périodiquement entre $r_{min} = 1.7 D/(2\pi)$ et $r_{max} = 3.3 D/(2\pi)$ avec une fréquence $f = 1\text{ Hz}$. L'extrémité de la méduse est également légèrement modifiée à la fréquence $f = 1\text{ Hz}$. Cette déformation ne préserve pas le volume. Une homothétie est alors imposée pour conserver au mieux le volume de la lentille non déformée (une erreur de 3% est observée). Ces paramètres de nage (r_{min} , r_{max} et f) sont choisis pour mimer une vraie méduse. La déformation du profil de ce prototype de méduse durant une période de nage est représenté sur la figure 3.25. La densité du fluide et du corps est $\rho = 1000\text{ kg/m}^3$, et la viscosité dynamique de l'eau est $\mu = 10^{-3}\text{ Pa}\cdot\text{s}$.

Le centre de masse de la méduse est initialement placé en $(x, y, z) = (0, 0, 0)$. La méduse se déplace grâce aux efforts exercés par le fluide sur le corps. La méduse nage dans le sens positif x dans un aquarium de dimension $[-2D, 6D] \times [-2D, 2D] \times [-2D, 2D]$. Le maillage est composé de $600 \times 300 \times 300 = 54$ millions de nœuds. Nous considérons un fluide au repos ($\mathbf{u} = \mathbf{0}$ à $t = 0$) avec des conditions aux limites de Dirichlet homogènes ($\mathbf{u} = \mathbf{0}$).

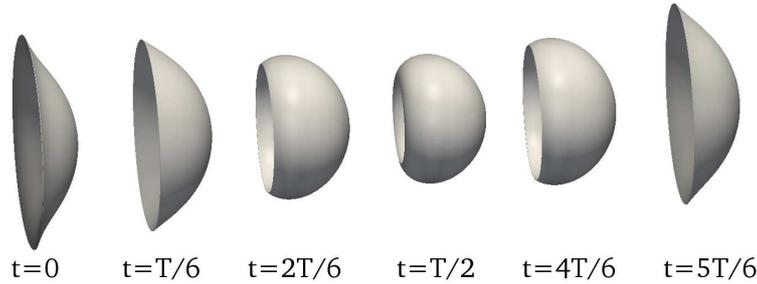


FIGURE 3.25 – Déformation de la méduse pendant une période de nage.

Des erreurs numériques qui affectent la conservation de masse peuvent se produire près des frontières mobiles (interface fluide/structure) lorsque des points de maillage rentrent et surtout sortent du corps immergé (Lee *et al.*, 2011). La figure 3.27 représente l'évolution temporelle des forces agissant sur le corps. Nous observons que les oscillations sont relativement faibles dans notre simulation grâce au fait que nous avons choisi une condition CFL proche 1 (Lee *et al.*, 2011). Le champ de pression dans un plan 2D est représenté sur la figure 3.28. La pression est continue à travers l'interface. La méthode IPC étant utilisée, aucune condition aux limites pour la pression n'est imposée sur l'interface (grâce à l'étape de pénalisation). Le champ de divergence de l'étape de prédiction est représenté sur la figure 3.29. La divergence est de l'ordre des résidus du problème linéaire à résoudre pour la projection. De plus, l'erreur sur la divergence est distribuée dans tout le domaine et pas nécessairement proche de l'interface.

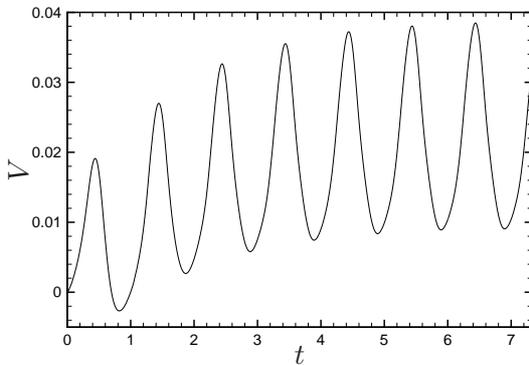


FIGURE 3.26 – Evolution temporelle de la vitesse de la méduse.

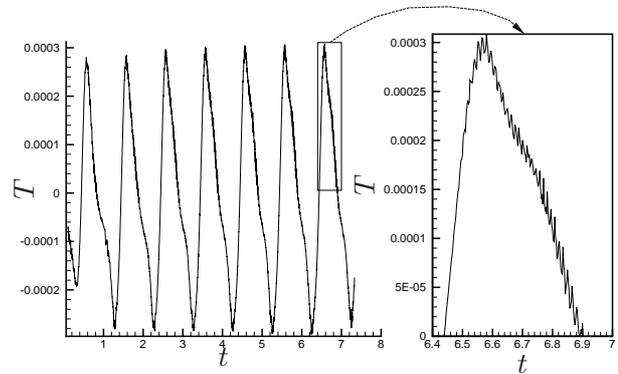


FIGURE 3.27 – Evolution temporelle de la force générée par la méduse.

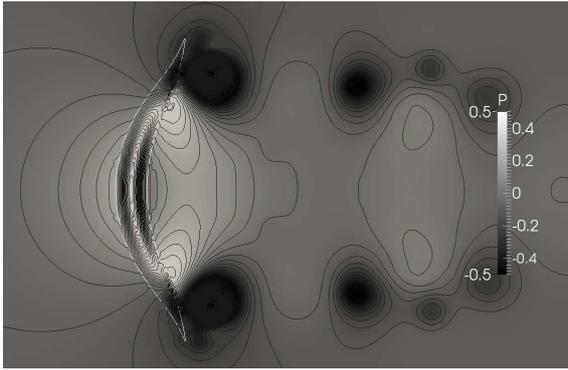


FIGURE 3.28 – Champ de pression généré par une méduse ($z = 0$).

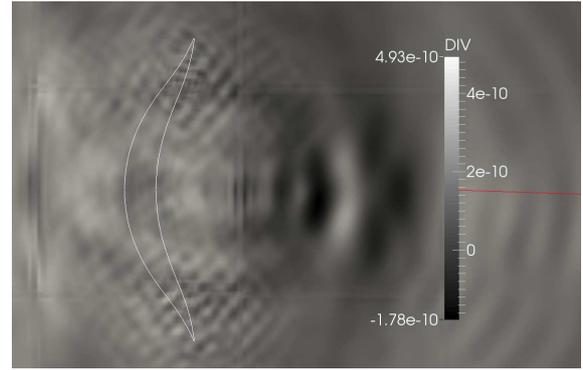


FIGURE 3.29 – Divergence du champ de vitesse généré par une méduse ($z = 0$).

Après un régime transitoire, la méduse atteint une vitesse aux alentours de 2.5 cm/s (figure 3.26). Le nombre de Reynolds associé est $Re = \frac{\rho V D}{\mu} \approx 1000$.

Le sillage généré par la méduse est représenté sur la figure 3.30. Des allées tourbillonnaires contrarotatives sont observées (voir figure 3.31). En raison du profil axisymétrique du corps et de la nage, ces allées tourbillonnaires sont similaires à celles générées par un prototype 2D (Bergmann et Iollo, 2011).

Les structures cohérentes sont semblables à celles observées expérimentalement par Dabiri *et al.* (2005) avec une analyse PIV du champ de l'écoulement généré par une vraie méduse (oblate medusian jellyfish, *Aurelia Aurita*) dans son habitat naturel. Bien que la nage semble réaliste, des comparaisons quantitatives sont difficilement réalisables par manque d'informations sur la vraie méduse. Les caractéristiques d'écoulement sont tout de même en très bon accord avec celles observées dans la nature.

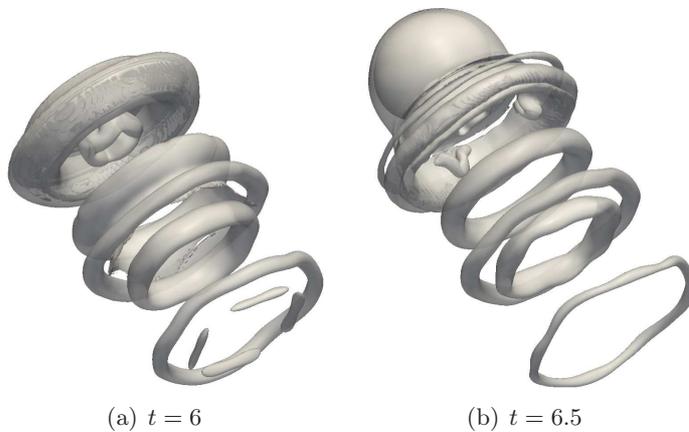


FIGURE 3.30 – Iso valeurs de vorticité du sillage tridimensionnel généré par une méduse.

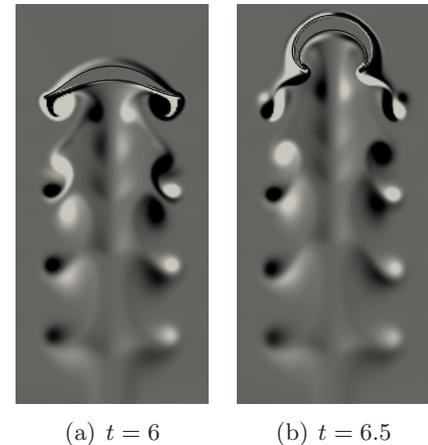


FIGURE 3.31 – Coupe bidimensionnelle du sillage généré par une méduse.

Raie

Ce petit paragraphe présente brièvement des simulations numériques de la nage d'un prototype de raie. Le but actuel n'est pas d'étudier en détail la physique de l'écoulement, mais d'illustrer encore un peu plus la possibilité des méthodes développées à simuler des écoulements autour de géométries complexes. Le profil stationnaire de la raie est présenté sur la figure 3.32. Ce profil est construit à l'aide de B-Splines. Le profil de nage est obtenu en déformant le profil stationnaire avec des lois de nage qui se rapprochent de celles observées dans la nature (voir figure 3.33). Le sillage généré par ce prototype de raie est représenté sur la figure 3.34. Nous voyons apparaître distinctement deux allées tourbillonnaires, chacune générée par une nageoire latérale de la raie (figure 3.34(a)). Enfin, chaque allée tourbillonnaire est finalement très similaire à celle générées par un prototype de poisson (figure 3.34(b)), avec une forme caractéristique en "V".

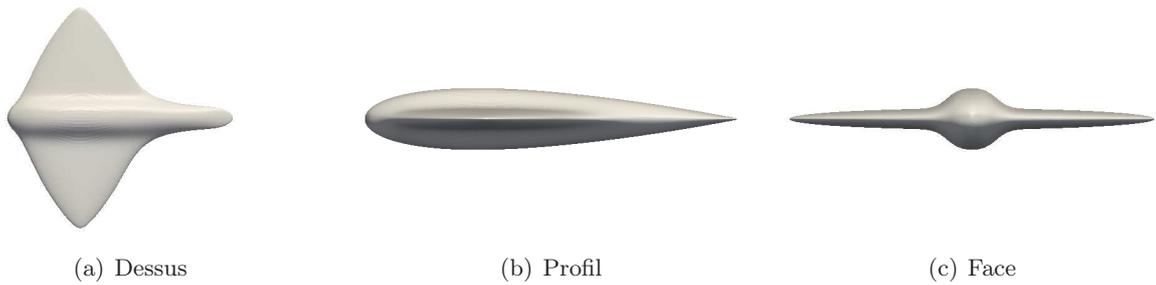
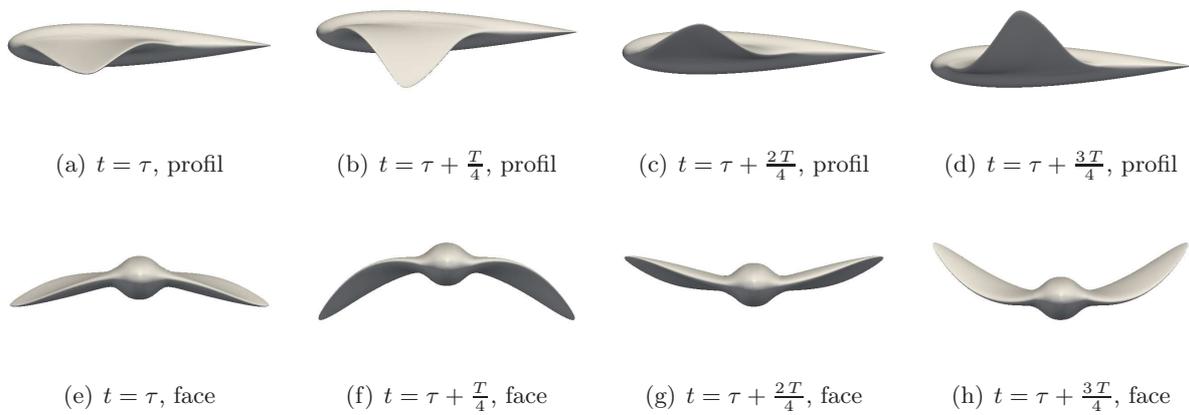


FIGURE 3.32 – Profil d'un poisson de type raie.

FIGURE 3.33 – Nage d'un poisson de type raie sur une période de battements T .

Knifefish

La géométrie du poisson knifefish est relativement proche de celle de la raie (figure 3.35). De même, le profil de nage est similaire (figure 3.36). Cependant, les amplitudes, les fréquences et les longueurs d'ondes de la loi de nage sont différentes. Le sillage généré par le knifefish est un peu différent de celui de la raie dans la mesure où les deux allées tourbillonnaires (qui existent même si elles sont moins visibles, voir figure 3.37(a)) n'ont pas la forme caractéristique en "V" (figure 3.37(b)).

3.4.6 Simulations numériques de la nage d'un poisson à partir d'images

Le but de cette section est de réaliser des simulations numériques à partir d'une séquence de la géométrie d'un corps (par exemple le profil de nage d'un poisson). Ces géométries peuvent être obtenues de différentes manières. En particulier, nous allons nous intéresser à la nage d'un poisson réaliste obtenu à partir de photographies. Le chemin pour parvenir à une simulation numérique 3D à partir d'images 2D est le suivant. Tout d'abord nous allons reconstruire le profil 3D du poisson à partir d'images 2D. Pour ce faire, une technique de squelettisation est utilisée. À partir du squelette du poisson (que nous pouvons déformer comme le ferai un vrai poisson) nous pouvons reconstruire une fonction level set associée à la géométrie 3D (moyennant quelques hypothèses assez réalistes). Si l'on se donne une loi de nage, cette technique peut être effectuée pour différents instants et nous donne donc une série de snapshots de fonctions level set de la géométrie 3D de poisson déformé. Comme nous l'avons vu précédemment, la simulation numérique nécessite la connaissance de la level set à chaque pas de temps de la simulation mais également du champ de vitesse dans le corps pour effectuer la pénalisation (ou frontière immergée). En général, nous avons uniquement un nombre limité de snapshots de la level set (en raison des coûts CPU) et il est donc nécessaire de reconstruire des snapshots intermédiaires. En résumé nous avons maintenant besoin de snapshots intermédiaires ainsi que le champ de déplacement (vitesse) entre deux snapshots successifs. Pour ce faire, le transport optimal répond parfaitement

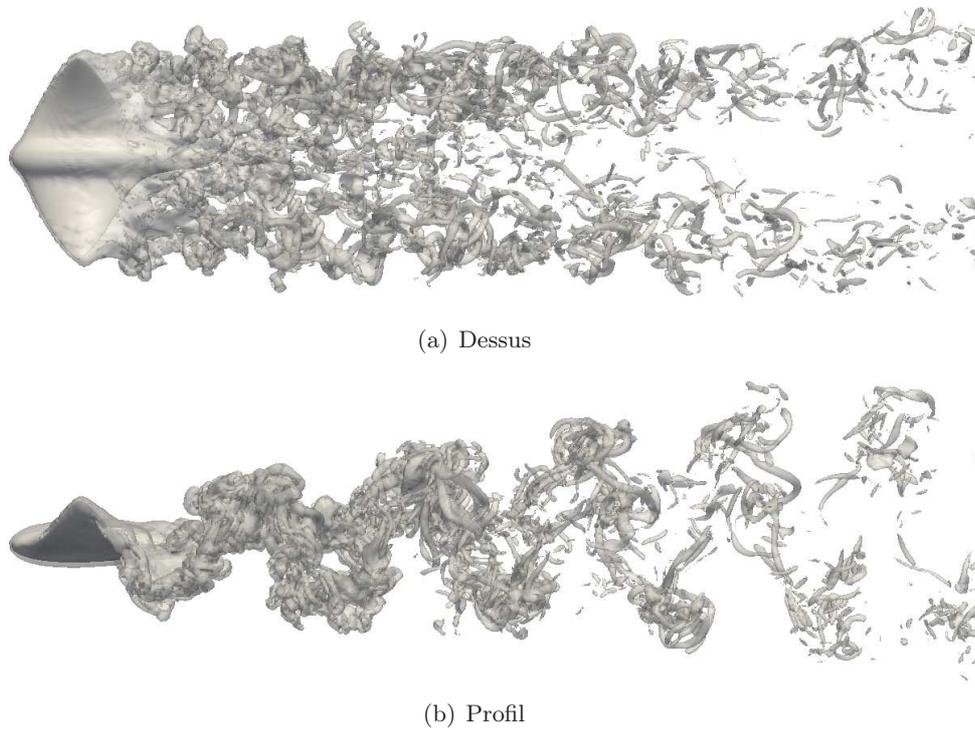


FIGURE 3.34 – Sillage généré par la nage d'un poisson de type raie (iso-vorticité).

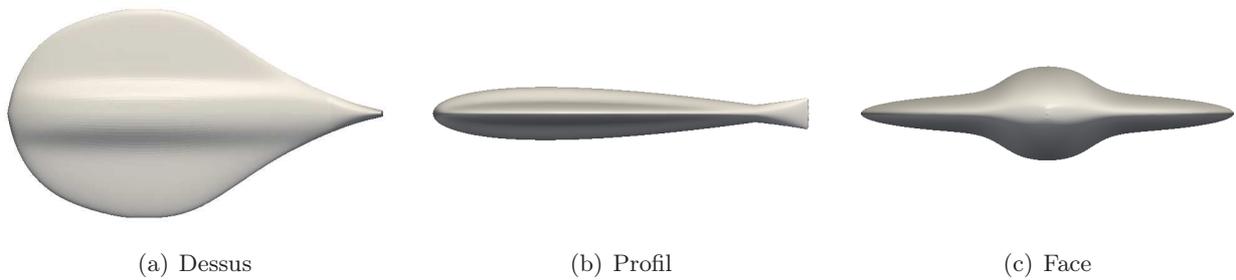
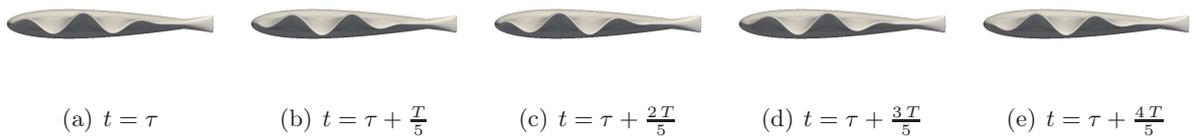


FIGURE 3.35 – Profil d'un poisson de type raie.

FIGURE 3.36 – Nage d'un poisson de type raie sur une période de battements T .

à cette problématique. A partir du moment où la level set est connue à chaque pas de temps de la simulation, ainsi que le champ de vitesse correspondant, nous pouvons effectuer la simulation numérique.

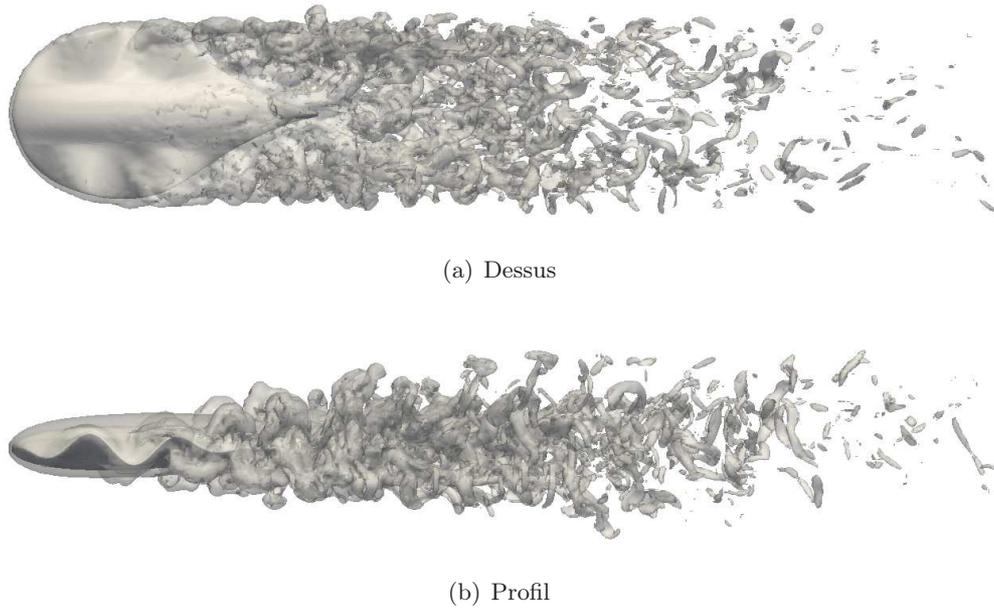


FIGURE 3.37 – Sillage généré par la nage d’un poisson de type knifefish (iso vorticité).

Méthode de squelettisation et de reconstruction 3D du profil

La méthode de squelettisation étudiée dans cette section a été développée par Cédric Galusinski (IMATH, Université du Sud Toulon Var) et nous avons utilisé le code de calcul développé par Christian Nguyen (IMATH, Université du Sud Toulon Var) pour la segmentation des photographies couplé au code de Cédric pour la squelettisation et la construction de la fonction level set. Cette collaboration a été initiée lors de la thèse de Jessica Hovnanian qui a largement contribué à la squelettisation. Tous les détails techniques sont décrits dans la thèse de Jessica (Hovnanian, 2012), et nous présentons ici uniquement quelques étapes importantes. Comme le montre la figure 3.38, nous commençons à partir de photographies d’un maquereau² de profil, de dessus et de face. Nous réalisons ensuite une segmentation pour extraire le contour de chaque image. À partir de ces contours nous pouvons construire les fonctions level set. Au besoin, les vues de dessus et de face seront redressées, comme illustrée sur la figure 3.39 pour la vue de dessus. Une symétrie est nécessaire, et c’est pour cela que les nageoires latérales ont également été supprimées. Le profil non déformé est représenté comme l’iso contour zéro d’une fonction level set (voir figure 3.40). Le profil de nage déformé peut être obtenu de la même manière en appliquant la loi de nage (3.7) au squelette (ligne médiane) du profil avec vue de dessus (figure 3.39). Nous pouvons de la sorte générer tout une série de fonctions level set correspondant à différents instants t pour la loi de nage (3.7). Dans ce qui suit, nous considérons la nage périodique (3.7) et extrayons 20 snapshots de level set sur une période. Comme dans les sections précédentes nous avons recentré les fonctions level set de telles sortes que la déformation n’engendre pas de déplacement et rotation artificiels. Le centre de masse reste donc inchangé par l’application de la déformée.

Transport Optimal

Dans cette étude nous avons donc accès à 20 snapshots de la fonction level set $\{\psi_s^k\}_{k=0,\dots,19}$. Sans perte de généralité, la méthode est illustrée avec les quantités ψ_s^0 et ψ_s^1 . Nous souhaitons donc obtenir le champ de vitesse qui permet de passer de ψ_s^0 à ψ_s^1 et ainsi de reconstruire des champs intermédiaires ψ_s^ℓ pour $0 < \ell < 1$. Ce problème trouve de nombreuses applications dans diverses disciplines telles que la physique, la mécanique ou le traitement d’images (Villani, 2009). Cette problématique a été initialement abordée Monge en 1781. La résolution numérique de ce type de problème reste difficile et nous avons adopté la méthode lagrangienne introduite par Bouharguane *et al.* (2014).

Nous étudions la solution L^2 du problème de Monge-Kantorovich (MKP). Soit $\rho_0(\xi)$, $\rho_1(x)$ deux fonctions

2. Photographies réalisées par Jessica Hovnanian après un petit tour chez son poissonnier préféré.

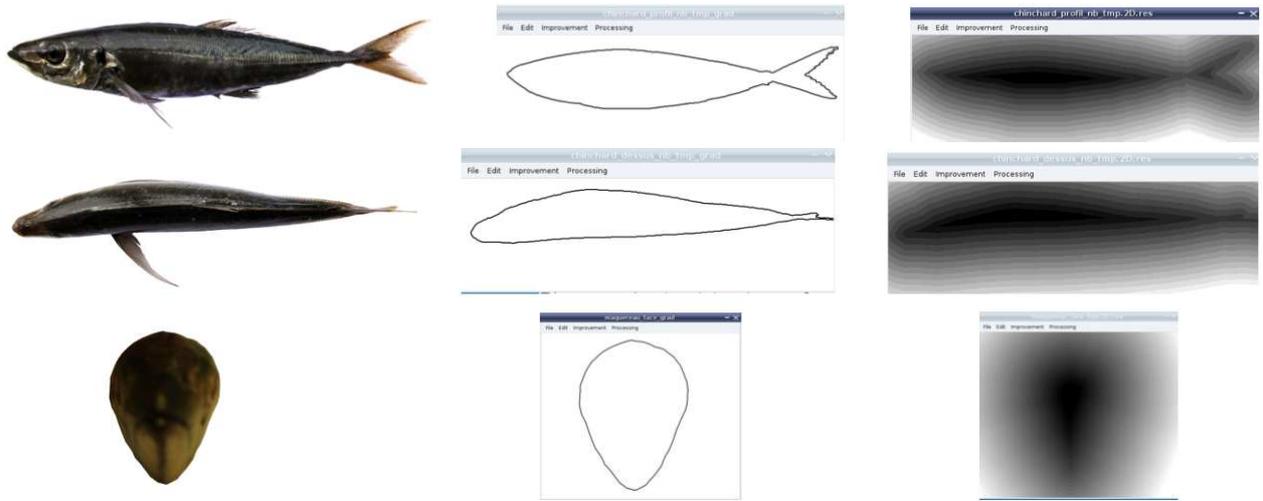


FIGURE 3.38 – Etapes de squelettisation : photos de profil et de dessus (gauche), segmentation et extraction du contour (milieu), construction de la level set (droite).



FIGURE 3.39 – Redressement du contour (vue de dessus).

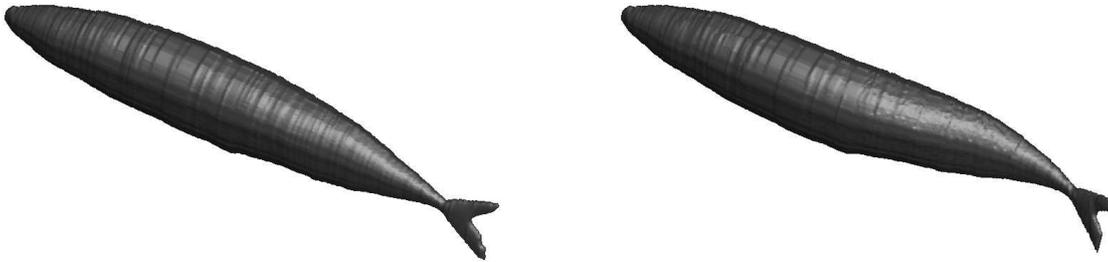


FIGURE 3.40 – Exemples de reconstruction du profil 3D. A gauche : profil non déformé. A droite : profil déformé.

scalaires non négatives à supports compacts Ω_0 et Ω_1 , où $\xi, x \in \mathbb{R}^d$ et d la dimension de l'espace ($d = 3$ dans notre étude). Nous supposons que

$$\int_{\Omega_0} \rho_0(\xi) d\xi = \int_{\Omega_1} \rho_1(x) dx.$$

Soit $X : \Omega_0 \rightarrow \Omega_1$ une transformation régulière telle que $X(\xi)$ est un transfert entre ρ_0 et ρ_1 . Nous avons alors l'équation Jacobienne :

$$\rho_0(\xi) = \det(\nabla X(\xi)) \rho_1(X(\xi)).$$

Cette équation est sous déterminée en fonction $X(\xi)$ et nous choisissons la transformation qui introduit la distance L^2 de Kantorovich-Wasserstein :

$$\inf_X \int_{\Omega_0} \rho_0(\xi) |X(\xi) - \xi|^2 d\xi. \quad (3.14)$$

Ce problème L^2 MKP correspond à déterminer la transformation X^* qui vérifie (3.14). Il a été démontré (Brenier, 1991; Villani, 2003, 2009) que ce problème admet une unique solution qui est le gradient d'une fonction convexe $\Psi : \Omega_0 \rightarrow \mathbb{R}$:

$$X^*(\xi) = \nabla \Psi(\xi).$$

La première classe de méthodes pour résoudre ce type de problème est basée sur la résolution de l'équation de Monge-Ampere (MAE) :

$$\rho_0(\xi) = \det(\nabla^2 \Psi(x)) \rho_1(\nabla \Psi(x)).$$

Le problème lié à cette équation est que les conditions aux limites restent inconnues.

La deuxième classe de méthodes est basée sur des arguments issus de la mécanique des milieux continus (Benamou et Brenier, 2000). Un temps fictif³ est introduit et nous avons alors $\Pi : [0, 1] \times \Omega_0 \rightarrow \mathbb{R}^d$, avec $\Pi(0, \xi) = \xi$, $\Pi(1, \xi) = X(\xi)$, $x = \Pi(t, \xi)$ et $\partial_t \Pi = v(t, x)$, et MKP est solution de

$$\inf_{\rho, v} \int_{\mathbb{R}^d} \rho(t, x) |v(t, x)|^2 dx,$$

où le minimum est recherché sur toutes les densités $\rho(t, x) \geq 0$ et les champs de vitesse $v(t, x) \in \mathbb{R}^d$ qui vérifient l'équation de continuité

$$\partial_t \rho + \nabla \cdot (\rho v) = 0, \quad (3.15)$$

avec les conditions initiales et finales :

$$\rho(0, \cdot) = \rho_0, \quad \rho(1, \cdot) = \rho_1.$$

La résolution de ce type de problème de minimisation avec contraintes fait intervenir des gradients et la résolution peut alors être très coûteuse en temps CPU. Nous allons utiliser la méthode lagrangienne développée par Bouharguane *et al.* (2014) basées sur des itérations de Picard. Dans cette méthode, nous devons calculer un champ de déplacement qui serait une perturbation du champ de déplacement optimal, et linéariser puis itérer autour de ce champ. Lorsque nous avons deux images assez proches l'une de l'autre, nous avons observé qu'il n'est pas nécessaire de calculer ce champ initial (le champ nul est acceptable). Nous avons donc initialement $\Psi^0 = 0$ et $X^0(\xi) = \xi$, où ξ représente les coordonnées du maillage fixe où sont définies les images. L'algorithme est alors le suivant :

Algorithme lagrangien pour la transport optimal : Pour une itération courante n , nous calculons l'image $\rho_0^n(\xi)$ de la quantité $\rho_1(X^n(\xi))$ par la transformation $X^n(\xi)$

$$\rho_0^n(\xi) = \rho_1(X^n(\xi)) \det \nabla_\xi X^n(\xi).$$

Idéalement, on souhaite au final $\rho_0^n(\xi) = \rho_0(\xi)$. A la première itération $n = 0$, nous avons $\rho_0^n(\xi) = \rho_1(X^n(\xi))$. Nous définissons un critère de convergence avec une certaine norme $|\rho_0^n(\xi) - \rho_0(\xi)| \leq \epsilon$. Si ce critère n'est pas vérifié, on calcule une correction Ψ^n en résolvant⁴ une approximation de l'équation (3.15) :

$$\rho_0^n(\xi) - \rho_0(\xi) = \nabla_\xi \cdot (\rho_1(X^n(\xi)) \nabla_\xi \Psi^n), \quad (3.16)$$

puis on met à jour la transformation

$$X^{n+1} = X^n - \alpha \nabla \Psi^n.$$

Le paramètre $0 \leq \alpha \leq 1$ dépend des images étudiées. Plus les images sont proches et plus ce paramètre peut tendre vers 1. Dans notre cas nous prenons $\alpha = 0.5$. Nous posons $n = n + 1$ et nous recommençons tant que le critère de convergence n'est pas vérifié. Soit $X^{opt} = X^n - \xi$ la solution finale. Afin que le problème (3.16) soit bien posé, nous régularisons au préalable les densités ρ_0 et ρ_1 qui sont nulles sur une partie du domaine de calcul en leur ajoutant une faible valeur ϵ . Cette valeur dépend des images et des maillages utilisés.

3. Qui ne sera pas fictif dans notre formulation, mais correspondra au pas de temps qui sépare deux images.

4. A ce stade nous considérons que le pas de temps entre deux images est égal à un. Nous corrigerons cela à la fin de l'algorithme.

Choix des densités : Les fonctions level set ne sont pas de bonnes candidates pour le transport optimal car nous devons avoir des densités strictement positives pour tout points de définition de l'image. Nous avons testé un masque (Heaviside des level set). Ce choix ne s'est pas avéré judicieux. En effet, la masse est essentiellement répartie sur le corps du poisson, mais très peu sur la nageoire caudale. Le transport au niveau de la nageoire n'est "pas jugé prioritaire" par l'algorithme. Nous avons alors considéré une gaussienne qui a approximativement un support égale à $\ell/10$ (ℓ est la longueur du poisson) de valeur maximale unité et localisée proche de l'interface de telle sorte que tout le profil du poisson ait le même poids dans l'algorithme. Nous avons régularisé les densités en leur ajoutant $\varepsilon = 10^{-4}$.

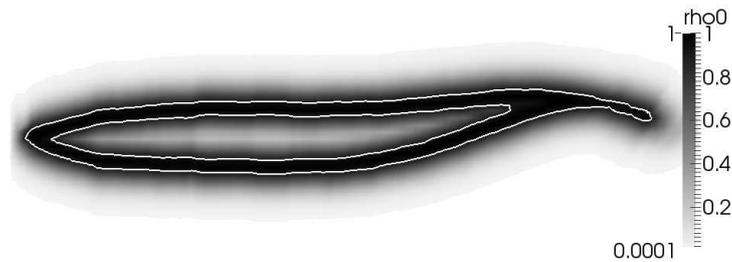


FIGURE 3.41 – Visualisation de la densité utilisée. Le zéro de la level set correspond au trait blanc extérieur. Coupe 2D, plan $z = 0$.

Vitesse de déformation : Avec un pas de temps Δt_s entre les deux images, le champ de vitesse de déformation qui permet de passer de l'image ρ_0 à l'image ρ_1 est $\tilde{\mathbf{u}}^0 = X^{opt}/\Delta t_s$. L'exposant 0 souligne le fait que cette vitesse de déformation agit sur des images de poisson dont le centre de masse est fixe en espace.

Construction d'images intermédiaires : La construction d'images intermédiaires est alors très simple. Pour $X = X^{opt} + \xi$ nous avons ρ_0 et pour $X = \xi$ nous avons ρ_1 . Soit τ un paramètre qui traduit le pourcentage d'avancement de ρ_0 vers ρ_1 . La transformation qui permet d'obtenir une image pour un paramètre τ fixé est $X_\tau = \xi + (1 - \tau) X^{opt}$. L'image est alors ρ_τ calculée par

$$\rho_\tau(\xi) = \rho_1(X_\beta(\xi)) \det \nabla_\xi X_\beta(\xi).$$

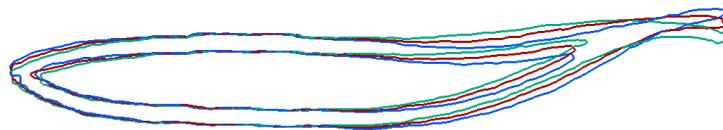


FIGURE 3.42 – Reconstruction d'une image pour $\tau = 0.5\Delta t_s$ (rouge) entre ρ_0 (vert) et ρ_1 (bleu). Coupe 2D, plan $z = 0$.

Généralisation : Nous venons de voir une méthode pour calculer un champ de vitesse de déformation d'une densité ρ_0 à une densité ρ_1 et ainsi de reconstruire des densités intermédiaires ρ_τ . Ceci peut être généralisé à toute une série de densités $\{\rho_i\}_{i=0}^{N_i}$ en calculant les vitesses des densités ρ_i vers les densités ρ_{i+1} . A partir des densités $\rho_\tau(\xi, t)$ nous pouvons calculer un masque (fonction caractéristique) et au besoin une fonction level set par redistanciation. Dans la suite nous notons $\psi^0(\xi, t)$ les fonctions level set des profils déformés et $\tilde{\mathbf{u}}^0(\xi, t)$ la vitesse de déformation.

Simulation numérique de la nage

Calcul de la position et de la vitesse du poisson : La méthode numérique de résolution est la même que dans les sections précédentes. Puisque nous nous intéressons à des corps autopropulsés, la position de corps sera représentée par une fonction level set $\psi_s(\xi, t)$ image de la fonction level set $\psi^0(\xi, t)$ par la transformation qui caractérise le mouvement rigide (translation plus rotation), notée $X^r(\xi, t)$. Nous avons donc :

$$\psi_s(\xi, t) = \psi^0(X^r(\xi, t)).$$

De même, la vitesse du corps est $\mathbf{u}^0(\xi, t) = \bar{\mathbf{u}}(\xi, t) + \mathbf{u}^\theta(\xi, t) + \tilde{\mathbf{u}}^0(\xi, t)$. Puisque la position du corps a été modifiée, cette vitesse doit également agir sur la position modifiée, et nous avons finalement :

$$\hat{\mathbf{u}}(\xi, t) = \mathbf{u}^0(X^r(\xi, t)).$$

La transformation $X^r(\xi)$, ainsi que les vitesses de translation $\bar{\mathbf{u}}(t)$ et de rotation $\mathbf{u}^\theta(\xi, t)$ sont évaluées par les lois de Newton en calculant les forces et les moments exercés par le fluide sur l'obstacle.

Calcul des forces et moments : Dans cette section la géométrie du poisson est défini de façon implicite comme étant le zéro de la fonction level set que nous venons de décrire. En l'absence de marqueurs sur le profil du poisson, nous ne pouvons pas calculer directement les forces et les moments à partir des relations (3.1a) et (3.1b). Nous considérons alors un domaine arbitraire Ω_{f_i} qui contient uniquement l'obstacle Ω_i (voir figure 3.43) de telle sorte que les forces s'écrivent :

$$\begin{aligned} \mathbf{F}_i = & -\frac{d}{dt} \int_{\Omega_{f_i}(t)} \mathbf{u} \, dV + \int_{\partial\Omega_{f_i}(t)} (\mathbb{T} + (\mathbf{u} - \mathbf{u}_i) \otimes \mathbf{u}) \mathbf{n}_i \, dS \\ & - \int_{\partial\Omega_i(t)} ((\mathbf{u} - \mathbf{u}_i) \otimes \mathbf{u}) \mathbf{n}_i \, dS. \end{aligned} \quad (3.17a)$$

De façon similaire, les moments s'écrivent :

$$\begin{aligned} \mathcal{M}_i = & -\frac{d}{dt} \int_{\Omega_{f_i}(t)} \mathbf{r}_i \wedge \mathbf{u} \, dV + \int_{\partial\Omega_{f_i}(t)} \mathbf{r}_i \wedge (\mathbb{T} + (\mathbf{u} - \mathbf{u}_i) \otimes \mathbf{u}) \mathbf{n}_i \, dS \\ & - \int_{\partial\Omega_i(t)} \mathbf{r}_i \wedge ((\mathbf{u} - \mathbf{u}_i) \otimes \mathbf{u}) \mathbf{n}_i \, dS. \end{aligned} \quad (3.17b)$$

Remarque : les intégrales sur $\partial\Omega_i$ dans les équations (3.17a) et (3.17b) s'annulent dans la plupart des applications ($\mathbf{u} = \mathbf{u}_i$), excepté le cas des frontières transpirantes (soufflage/aspiration par exemple). Dans ce qui suit, pour des raisons de simplicité de mise en œuvre, les volumes de contrôle Ω_{f_i} sont des rectangles qui entourent les obstacles et qui coïncident avec les nœuds du maillage fluide.

Validation : nous comparant ici deux simulations numériques où la géométrie du poisson est calculée par B-spline et est donc connue à chaque instant :

- Cas *I* ("Lagrangien") : la fonction level set et la vitesse de déformation sont données à chaque pas de temps de la simulation (comme dans les premières sections de ce chapitre)
- Cas *II* ("Eulérien") : nous supposons que l'on a uniquement à disposition 20 snapshots de la géométrie sur une période, et nous devons alors calculer les vitesses de déformation et les géométries manquantes (à chaque pas de temps de la simulation) par transport optimal.

Dans les deux cas la taille du poisson est fixée à $\ell = 10 \text{ cm}$ et une fréquence de nage $f = 4 \text{ Hz}$ est utilisée avec $c_0 = -0.002$, $c_1 = -0.12$, $c_2 = 2$ et $\lambda = \ell$ dans la loi de nage (3.7). Ces paramètres sont proches de ceux observés dans la nature (Yu *et al.*, 2011). La figure 3.44 montre une comparaison du champ de vitesse obtenu dans le cas *I* (suivi de marqueurs sur le poisson) avec le champ de vitesse obtenu dans le cas *II* (transport optimal) dans une coupe 2D. Nous représentons les composantes U (suivant la direction de nage du poisson) et la composante W (dans une direction normale à la nage). La loi de nage est définie dans le plan (X, Z) . Une très bonne concordance est obtenue pour la composante W . Une concordance moins évidente est présente pour la composante U . De part la géométrie du profil, nous pouvons considérer que la vitesse U est approximativement la vitesse tangentielle au profil. Le transport optimal n'étant pas en mesure (avec uniquement des informations géométriques) de modéliser les rotations par exemple, et les vitesses tangentielles

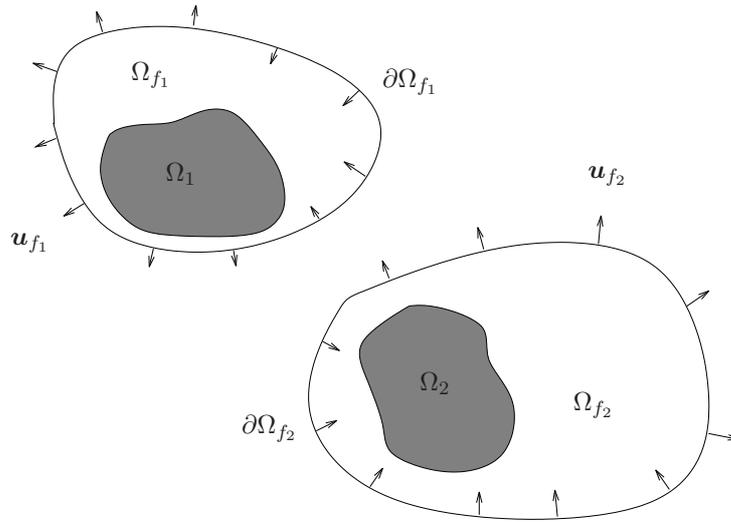
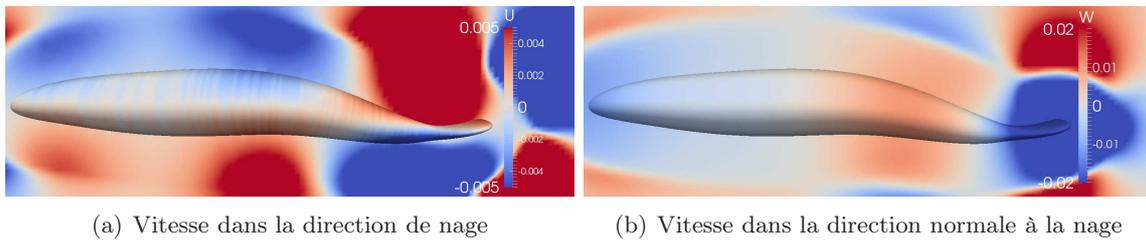


FIGURE 3.43 – Représentation schématique des domaines utilisés pour le calcul des forces et moments.

ne peuvent donc pas être calculées avec ce type de méthode. Il faudrait par exemple avoir des informations supplémentaires telles que la répartition de la densité dans le poisson. Une autre solution pour remédier à ce problème serait d'ajouter *a priori* dans la formulation du problème du transport optimal, une contrainte de conservation de la masse pour le champ de vitesse obtenu. Les vitesses de nage obtenues dans les deux



(a) Vitesse dans la direction de nage

(b) Vitesse dans la direction normale à la nage

FIGURE 3.44 – Comparaison des vitesses obtenues dans le cas *I* (suivi Lagrangien, champ dans le poisson) et dans le cas *II* (transport optimal, représentée uniquement à l'extérieur du corps).

situations sont comparées sur la figure 3.45. La vitesse normale W présente une très bonne concordance pour les deux méthodes. La vitesse de nage U est plus influencée par les différences sur la composante tangentielle de vitesse observée sur la figure 3.44. Néanmoins, malgré cette différence, la vitesse de nage est en bon accord avec la vitesse de référence obtenue par un suivi Lagrangien. Après cette validation, nous allons effectuer une simulation numérique à partir d'images d'un maquereau dont nous n'avons pas accès à un suivi Lagrangien (pas de maillage du poisson).

Résultats à partir de photos d'un maquereau : La figure 3.46 présente le sillage généré par le poisson reconstruit à partir d'images. Les paramètres utilisés sont les mêmes que précédemment, à savoir que la taille du poisson est fixée à 10 cm et une fréquence de nage $f = 4\text{ Hz}$. La vitesse de nage du maquereau est supérieure à celles obtenues dans les deux cas précédents ($U \approx -0.15$). Cette augmentation peut s'expliquer par un profil du corps différent, mais également par la forme en croissant de lune de la nageoire caudale. Des analyses plus détaillées du sillage sont en cours.

3.5 Conclusions et perspectives

Ce chapitre avait pour objectif initial d'illustrer l'efficacité des méthodes utilisées pour simuler des obstacles qui se déforment et se déplacent dans un fluide. Un prototype de nageur était de ce fait un bon candidat

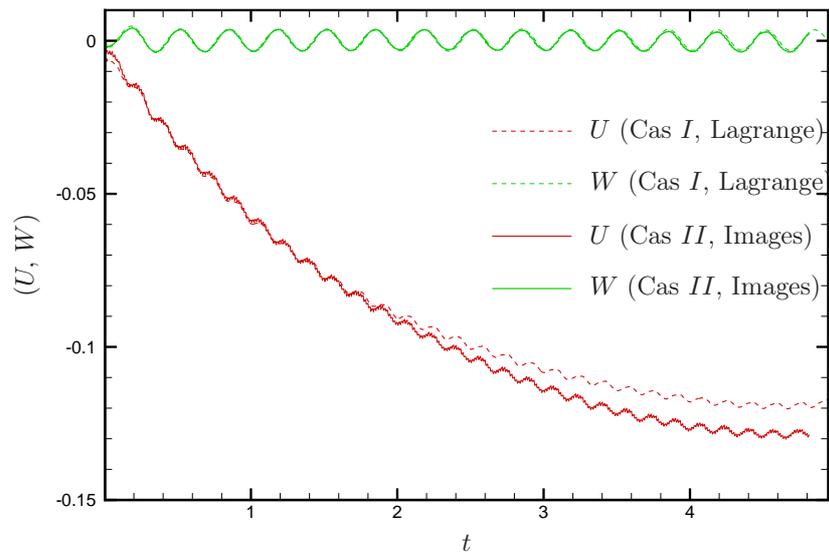


FIGURE 3.45 – Comparaison des vitesses obtenues par le poisson "lagrangien" et le poisson reconstruit à partir d'images.

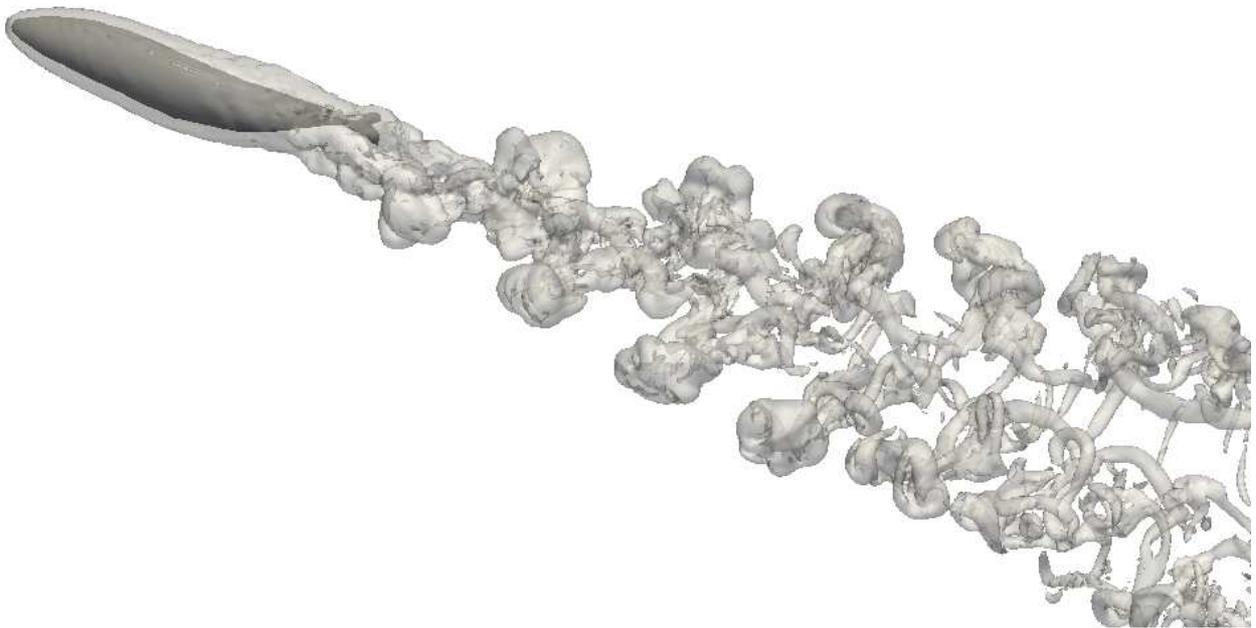


FIGURE 3.46 – Simulation de la nage d'un maquereau reconstruit à partir de photographies.

pour illustrer l'efficacité de ces méthodes. Finalement, nous avons décidé d'étudier plus profondément les mécanismes de la nage en deux et trois dimensions. En particulier nous avons observé que toutes les nages ne sont pas efficaces d'un point de vue énergétique. Par exemple, nous avons illustré l'importance de la flexibilité de la nageoire caudale, à la façon d'une palme pour un plongeur. Enfin, nous avons développé une stratégie pour simuler la nage d'un poisson dont la géométrie est obtenue par des photographies. Cette méthode couple squelettisation, transport optimal et simulations numériques directes. En dehors d'essayer d'améliorer le transport optimal pour les déformations tangentielles à la paroi, ce chapitre est terminé et ne présente pas

de réelles perspectives à court terme sur la nage. Par contre les perspectives énergétiques sont nombreuses, et sont reportées dans le chapitre suivant avec l'étude de systèmes de récupération de l'énergie, tels que les éoliennes et les serpents de mers pour extraire de l'énergie du vent et de la houle.

Finalement, les méthodes que nous avons utilisées et développées, nous permettent de réaliser des simulations qui seraient relativement complexes avec d'autres outils de simulation. La simulation est très rapide car elle est massivement parallèle de part la structure cartésienne du maillage. La méthode numérique est très efficace car elle est basée sur l'utilisation de schémas aux différences finies qui sont simples, précis et fiables. Un point essentiel également est la mise en route très rapide des simulations. Toutes les simulations présentées dans ce chapitre ont été effectuées avec le même code de calcul avec une mise en route immédiate, quelle que soit la complexité de la géométrie étudiée. Bien sûr, il a fallu imaginer la forme des structures et leurs déformation. Mais le code de calcul NaSCar permet de lire n'importe quelle géométrie (maillage) et calcule une fonction level set associée. A partir de ce moment, la simulation peut démarrer. Même si la méthode peut être, dans certains cas, moins efficace que le serait une méthode avec un maillage conforme (on peut avoir plus de points de maillage), elle permet de s'affranchir du temps "humain" pour la mise en route des simulations. Ce temps "humain" est transféré en temps "machine".

Chapitre 4

Etude de quelques systèmes de récupération de l'énergie

Sommaire

4.1	Introduction	99
4.2	Modélisation et simulations numériques d'éoliennes	100
4.2.1	Description et contexte de l'étude : collaboration avec VALOREM	100
4.2.2	Modèle simplifié et code de calcul VALDAG	101
4.2.3	Simulations tridimensionnelles et limites de l'étude	105
4.3	Récupération de l'énergie de la houle	109
4.3.1	Un modèle de structure élastique	110
4.3.2	Résolution numérique et couplage	114
4.3.3	Modélisation et simulations numériques d'écoulements bifluïdes	116
4.3.4	Gestion de la ligne triple	117
4.3.5	Vers les serpents de mer	119
4.4	Conclusions et perspectives	120

4.1 Introduction

Ce chapitre prospectif a pour objectif d'entamer l'étude de quelques systèmes de récupération de l'énergie qui est présente dans le fluide environnement. Les méthodes et résultats présentés dans ce chapitre sont pour la plupart en cours de développement, et n'ont pas le même degré de maturité que ceux présentés dans les chapitres précédents. Nous avons toujours besoin de développer des techniques de modélisation et de simulations numériques efficaces pour aller vers les applications. Dans ce qui suit, nous nous sommes intéressés à deux applications particulières, qui sont encore en cours d'étude et de développement.

La première application est l'étude et l'optimisation de la forme de pale d'éoliennes pour maximiser la puissance à extraire d'un vent donné (§4.2). Cette étude s'inscrit dans une collaboration industrielle avec le groupe VALOREM (producteur d'énergie verte, <http://www.valorem-energie.com>) et de sa filiale VALEOL (<http://www.valeol.fr>) localisée à Bègles en Gironde. Cette collaboration a été initiée en 2010 conjointement avec Angelo Iollo (IMB), Bastien Gaillardon et Serge Gracia (VALEOL) et a conduit à une première thèse cife (Xin Jin, soutenance prévue en septembre 2014) qui devrait être suivie d'une seconde qui démarrerait en septembre 2014.

La seconde application est liée à des écoulements bifluïdes. Nous envisageons de modéliser et simuler des structures articulées ou élastiques posées sur la surface de l'océan et capables d'extraire de l'énergie de la houle (§4.3). Ce travail est effectué en collaboration avec Lisl Weynans (IMB) sur la modélisation des écoulements bifluïdes et notamment la résolution de problèmes elliptiques à coefficients discontinus, ainsi qu'avec Franky Luddens (post doc dans l'équipe) sur le calcul de fonctions level set précises (pour une bonne conservation du volume et un calcul de la courbure précis). Ce travail s'effectue également avec d'étroites discussions avec Pierre Lubin (I2M, Bordeaux) qui est spécialiste de l'étude de la houle déferlante. Avant

d’arriver à simuler totalement un serpent de mer, des étapes en amont doivent être menées. Nous allons proposer un modèle et une méthode numérique pour résoudre des écoulements bifluïdes (§4.3.3). Puisque ce travail est en cours, nous étudierons quelques instabilités bi-fluïdes pour valider le modèle, puis nous nous intéresserons alors à l’étude d’une première interaction fluide/fluide/obstacle qui est le déplacement d’un bateau (§4.3.4). Enfin, nous présenterons les premiers développements pour un serpent de mer en deux dimensions (§4.3.5).

4.2 Modélisation et simulations numériques d’éoliennes

4.2.1 Description et contexte de l’étude : collaboration avec VALOREM

Les méthodes de modélisation et de simulations numériques développées au chapitre 2 sont particulièrement adaptées pour des obstacles en mouvement. Nous avons donc illustré ces méthodes pour la nage de poissons. Une autre application possible est l’écoulement autour d’un rotor. Nous avons étudié différents types de rotors, comme des hélices de moteurs pour bateaux par exemple (voir figure 4.1). Cette étude préliminaire

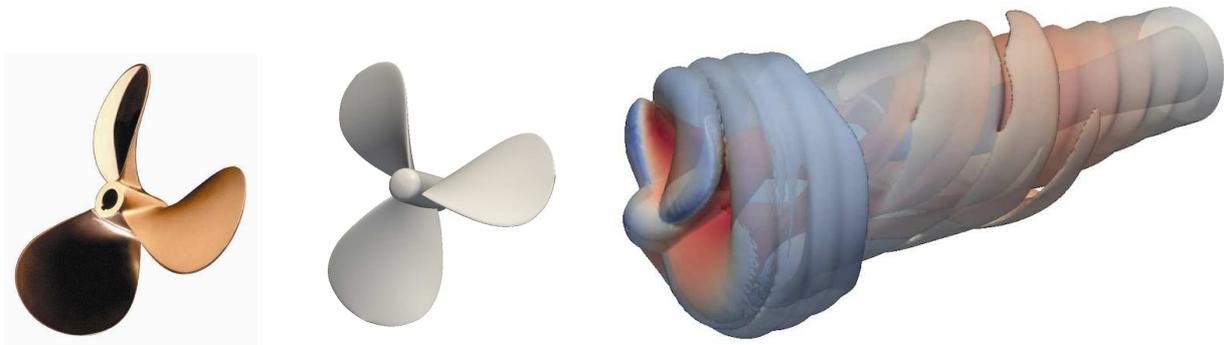


FIGURE 4.1 – Exemple de simulation d’une hélice de bateau. Photo d’une hélice (gauche), modélisation de l’hélice (centre) et simulation (droite, iso vorticité à $Re = 10^4$).

nous a amené à étudier des rotors d’éoliennes. Nous avons alors contacté le groupe VALOREM et plus particulièrement sa filière VALEOL (Bastien Gaillardon et Serge Gracia) qui s’intéresse à la conception d’éoliennes, et en particulier à l’optimisation de la forme des pales. Un des objectifs du projet est la quantification de l’apport d’appendices de pales en terme de puissance. Bastien et Serge se sont montrés très intéressés par le type de simulations que nous développons et une collaboration a commencé avec le stage de fin d’études puis la thèse cifre de Xin Jin que j’encadre avec Angelo Iollo. Les coûts de calculs nécessaires pour optimiser des pales d’éoliennes avec un calcul détaillé en trois dimensions sont cependant prohibitifs. Actuellement le problème nécessite la résolution de quelques centaines de millions d’inconnues et le temps CPU pour simuler deux tours du rotor est d’environ 4 jours sur 256 cœurs (300 Go de mémoire sont également nécessaires). Une optimisation de type adjoint n’est pas envisageable. L’idée est alors de réaliser une chaîne de conception basée sur un modèle simplifié pour l’optimisation et le modèle détaillé de Navier-Stokes en trois dimensions pour la validation. Tant que le modèle simplifié est jugé fiable, nous continuons le processus d’optimisation. Ceci est similaire à l’optimisation que j’avais effectué en thèse qui couplait optimisation par modèle réduit POD et modèle détaillé Navier-Stokes (Bergmann et Cordier, 2008).

Avant d’entamer le processus d’optimisation, nous avons étudié plusieurs types de pales, et nous étudions actuellement l’éolienne proposée par le NREL pour valider nos résultats numériques. En effet, il existe de nombreux résultats expérimentaux pour ce rotor obtenus dans la soufflerie de $24.4m \times 36.6m$ du NASA Ames¹ (figure 4.2). Ce rotor est constitué de deux pales d’environ $5m$ chacune et dont la géométrie est donnée sur la figure 4.2 et modélisée sur la figure 4.3. Le modèle simplifié est décrit à la section 4.2.2, et les simulations 3D sont présentées à la section 4.2.3

1. Cette soufflerie fait partie d’un grand complexe appelé National Full-Scale Aerodynamics Complex (NFAC) qui est implanté au NASA Ames Research Center à Moffett Field, (Silicon Valley) en Californie.



FIGURE 4.2 – Soufflerie et éolienne du NREL. <http://wind.nrel.gov/amestest/>

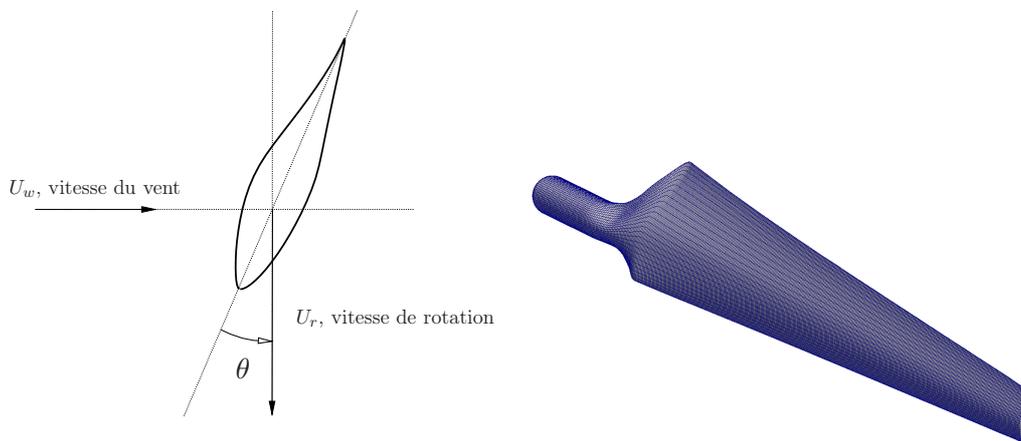


FIGURE 4.3 – Modélisation de la pale du NREL. Twist from 30° at hub to -2.5° at tip.

4.2.2 Modèle simplifié et code de calcul VALDAG

Le modèle simplifié étudié ici a été utilisé et développé dans le stage de fin d'étude et dans la thèse de Xin Jin. Les méthodes de simulation utilisées sont de type méthodes prédictives de rotor. Ces méthodes, basées sur le concept du disque actif, représentent un rotor par des forces équivalentes distribuées sur un disque perméable d'épaisseur nulle dans un domaine d'écoulement. Nous dénotons ce modèle comme simplifié car il est basé sur une hypothèse d'axisymétrie. La méthode générale est décrite par Sørensen (2011), et nous décrivons ici uniquement les principaux aspects qui nous ont amenés au modèle final. Notons que le modèle simplifié décrit dans la suite utilise les polaires (valeurs de traînée C_D et de portance C_L en fonction de l'angle d'incidence α pour un profil donné) pour calculer la puissance extraite par la turbine.

Description du modèle En comparaison avec le modèle classique de type BEM (*Blade Element Momentum*), le domaine étudié ici est un cylindre avec un rayon très grand par rapport à celui du rotor. Ce modèle est basé sur la théorie de disque actif généralisé, qui effectue une résolution numérique des équations de Navier-Stokes. La géométrie du rotor n'intervient pas dans cette formulation, mais seulement les forces équivalentes. En raison de l'axisymétrie du problème, le domaine de calcul est limité à un seul plan de symétrie

(r, z) (figure 4.4).

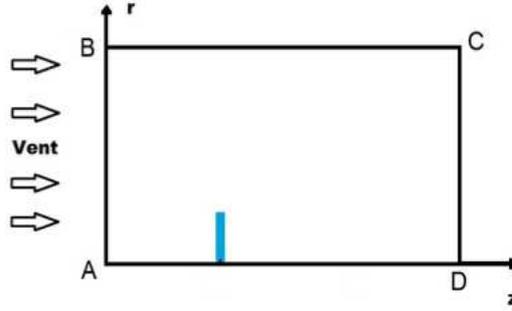


FIGURE 4.4 – Domaine de calcul 2D

La solution stationnaire des équations de Navier-Stokes nous donnera le champ de vitesse dans les trois directions (axiale, tangentielle et radiale) et un champ de forces d'interactions fluide/rotor et de la vorticit . Finalement, le but est de construire un diagramme de type $C_P(\lambda)$, o  C_P est le coefficient de puissance et λ un param tre appel  *type speed ratio* (ratio entre vitesse en bout de pale et vitesse du vent). Ce diagramme est tr s utile pour d termin er le point (λ) de fonctionnement optimal de l' olienne.

Equations Navier-Stokes en formulation $\Psi - \omega - V_\theta$ Pour les r gimes d' coulements  tudi s nous consid rons l'air comme un fluide incompressible. Les  quations de Navier-Stokes sont mises dans une formulation $\Psi - \omega - V_\theta$ en coordonn es cylindriques.

La fonction de courant Ψ est d finie par :

$$V_r = -\frac{1}{r} \frac{\partial \Psi}{\partial z} \quad \text{et} \quad V_z = \frac{1}{r} \frac{\partial \Psi}{\partial r} \quad (4.1)$$

qui satisfait l' quation de continuit  $\nabla \cdot \mathbf{V} = 0$. Les deux  quations de transport (pour la vorticit  ω et la vitesse tangentielle V_θ) s' crivent :

$$\frac{\partial \omega}{\partial t} + \frac{\partial(V_r \omega)}{\partial r} + \frac{\partial(V_z \omega)}{\partial z} - \frac{\partial}{\partial z} \left(\frac{V_\theta^2}{r} \right) = -\frac{\partial f_z}{\partial r} + \frac{1}{Re} \left(\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial(r\omega)}{\partial r} \right) + \frac{\partial^2 \omega}{\partial z^2} \right), \quad (4.2)$$

$$\frac{\partial V_\theta}{\partial t} + \frac{\partial(V_r V_\theta)}{\partial r} + \frac{\partial(V_z V_\theta)}{\partial z} + \frac{2V_r V_\theta}{r} = f_\theta + \frac{1}{Re} \left(\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial(rV_\theta)}{\partial r} \right) + \frac{\partial^2 V_\theta}{\partial z^2} \right), \quad (4.3)$$

o  le nombre de Reynolds $Re = V_0 R_0 / \nu$ est construit avec le rayon du rotor R_0 et la vitesse du vent V_0 .

R solution du champ de vitesse 2D Le champ de vitesse dans le plan (r, z) est d termin  par l' quation de Poisson pour la fonction de courant Ψ en coordonn es cylindriques :

$$\frac{\partial^2 \Psi}{\partial r^2} - \frac{1}{r} \frac{\partial \Psi}{\partial r} + \frac{\partial^2 \Psi}{\partial z^2} = -r\omega, \quad (4.4)$$

o  $\omega = \mathbf{e}_\theta \cdot (\nabla \times \mathbf{V})$ repr sente la composante du vecteur de vorticit  dans la direction \mathbf{e}_θ .

Les conditions aux limites sont :

– A l'entr e (AB) :

$$\Psi = \frac{V_0 r^2}{2}, \quad \omega = 0, \quad V_\theta = 0 \quad (4.5)$$

– A la sortie (CD) :

$$\frac{\partial \Psi}{\partial z} = 0, \quad \frac{\partial \omega}{\partial z} = 0, \quad \frac{\partial V_\theta}{\partial z} = 0 \quad (4.6)$$

– Sur l'axe de sym trie (AD) :

$$\Psi = 0, \quad \omega = 0, \quad V_\theta = 0 \quad (4.7)$$

– Sur le bord lat ral (BC) :

$$\Psi = \Psi_{AB}(r = Lr) = \frac{V_0 L r^2}{2}, \quad \omega = 0, \quad V_\theta = 0 \quad (4.8)$$

Calcul de forces aérodynamiques volumiques En supposant que le rotor a une vitesse de rotation Ω , on pose :

$$V_{rel}^2 = V_z^2 + (\Omega r - V_\theta)^2 \quad \phi = \tan^{-1} \left(\frac{V_z}{\Omega r - V_\theta} \right), \quad (4.9)$$

La force par unité de longueur radiale dr s'exprime :

$$\mathbf{F} = \frac{1}{2} \rho V_{rel}^2 c B C_F \quad (4.10)$$

où $\mathbf{F} = (F_z, F_\theta)^T$ et le coefficient de force C_F se déduit des coefficient de portance C_L et de traînée C_D . La force volumique est :

$$\mathbf{f}_{vol} = \frac{\mathbf{F} dr}{dV} = \frac{1}{2} \frac{\rho V_{rel}^2 c B C_F dr}{2\pi r dr k \Delta z} = \frac{\rho V_{rel}^2 c B C_F}{4\pi r k \Delta z}, \quad (4.11)$$

où B est le nombre de pales du rotor éolien, c la corde de la pale sur la section considérée. On en déduit donc les forces volumiques dans l'équation (4.1) :

$$f_{z_{vol}} = \frac{\rho V_{rel}^2 c B}{4\pi r k \Delta z} (C_L \cos \phi + C_D \sin \phi) \quad \text{et} \quad f_{\theta_{vol}} = \frac{\rho V_{rel}^2 c B}{4\pi r k \Delta z} (C_L \sin \phi - C_D \cos \phi). \quad (4.12)$$

A partir de ces forces nous pouvons facilement déduire le moment exercée sur le rotor et donc la puissance transmise par le vent.

Implémentation numérique Les équations sont discrétisées sur un maillage cartésien. On cherche une solution asymptotique, et le temps t dans les équations de transport représente un temps fictif pour atteindre la solution stationnaire. Les termes convectifs sont approchés par un schéma upwind et les autres termes par différence finies centrées. Un solveur linéaire implicite est utilisé pour résoudre chaque position radiale avec des positions à gauche et celles sur la même ligne. La figure 4.5 montre ce domaine de calcul avec ce solveur linéaire.

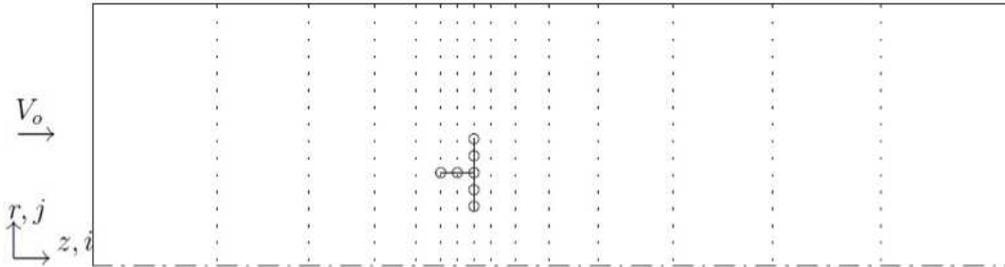


FIGURE 4.5 – Domaine de calcul avec un solveur linéaire implicite

L'utilisation de ce solveur linéaire permet de réduire considérablement les coûts de calcul. Par exemple, pour un maillage de $N_1 \times N_2$ (dans la direction radiale et axiale respectivement), au lieu de résoudre un système matriciel de dimension $N_1 \times N_2$, la dimension est N_1 avec une résolution répétée N_2 fois. Les systèmes linéaires sont résolus à l'aide d'une factorisation LU.

L'intégration en temps est réalisée avec des schémas de type Crank-Nicolson et Adams-Bashforth.

Corrections et calibrations Le modèle de VALDAG peut être amélioré par ajout de quelques corrections qui doivent permettre de prendre en compte des effets non résolus comme le *tip loss* qui est un effet purement tridimensionnel. Les simulations de rotors avec des modèles simplifiés de type VALDAG ou BEM montrent que pour des vitesses de vents supérieures à 13 m/s les modèles ne sont pas fiables car des phénomènes 3D non calculés deviennent trop importants. La prise en compte de ces effets est en cours d'étude. Nous nous sommes pour le moment limité à calibrer le modèle en minimisant l'erreur entre les puissances expérimentales et calculées pour des vitesses de vent $3 \text{ m/s} \leq V \leq 13 \text{ m/s}$. La calibration utilisée dans cette étude agit sur le twist pour différentes sections de la pale (nous calculons un nouveau $\phi(r) = \phi_0(r) + \phi'(r)$, où

ϕ_0 est le twist original et ϕ' la correction que l'on cherche). Afin de calibrer le modèle on souhaite minimiser la fonctionnelle

$$\mathcal{J}_1(\phi') = \int_{\mathcal{V}} (P_{exp} - P_{valdag}(\phi'))^2 dv + \alpha \|\phi'\|_2,$$

où \mathcal{V} représente un gamme de vitesses du vent. En pratique, nous nous sommes limités aux vitesses 5, 7, 10, 12 et 13 m/s. Le paramètre α permet de régulariser (ou de pénaliser) la fonction objectif afin d'avoir une correction de twist ϕ' bornée, et de préférence assez petite pour ne pas trop modifier la nature du système étudié. Cette calibration (optimisation) a été réalisée via un algorithme de simplexe de type Nelder-Mead (Nocedal et Wright, 1999). Le twist calibré est représenté sur la figure 4.6. On constate une réelle différence entre twist calibré et twist initial pour une abscisse proche de $r = 4$. Cette calibration limite l'effet de décrochage sur la partie du profil $r > 3$ (Jin *et al.*, 2012). La figure 4.7 présente une comparaison des

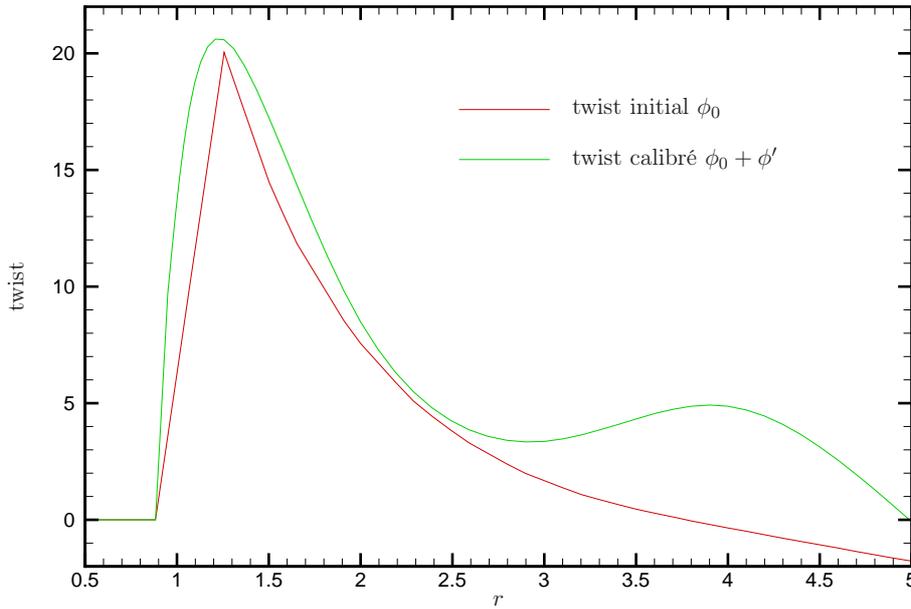


FIGURE 4.6 – Twist initial ϕ et twist modifié $\phi + \phi'$.

puissances obtenues en fonctions de la vitesse du vent pour les différentes approches considérées, à savoir BEM classique, VALDAG avec et sans calibration, avec les résultats expérimentaux. Nous voyons que les modèles BEM et VALDAG sans calibration donnent des résultats assez proches, mais la méthode BEM ne permet pas toujours d'obtenir facilement et rapidement une convergence (on itère entre théorie M (*Momentum*) et BE (*Blade Elements*), notamment pour les petites vitesses du vent. La méthode VALDAG avec calibration permet d'obtenir des résultats très proches des résultats expérimentaux.

Un première optimisation La société VALEOL n'est pas intéressée par la conception d'une nouvelle pale : en effet, il serait trop coûteux économiquement de modifier tout le parc éolien existant en l'équipant de nouvelles pales. VALEOL est plus intéressé par étudier le potentiel bénéfique d'ajout d'appendices sur des pales existantes. La première optimisation a été de calculer un angle d'attaque optimal (on veut modifier le twist sur toute la pale et donc calculer la constante ϕ'' telle que $\phi(r) = \phi_0(r) + \phi'(r) + \phi''$). La fonctionnelle à minimiser est alors

$$\mathcal{J}_2(\phi'') = \int_{\mathcal{V}} \frac{1}{P_{valdag}(\phi'')^2} dv + \beta \|\phi''\|_2.$$

Cette optimisation qui a également été réalisée via un algorithme de simplexe de type Nelder-Mead (Nocedal et Wright, 1999) permet de gagner 5% de puissance sur la plage de vents considérée avec $\phi'' = 3$ degrés. La seconde optimisation qui consiste à ajouter une partie profilée proche du pied de pale, comme les pales géantes développées par la société ENERCON (voir figure 4.8, <http://www.enercon.de/en-en/>). Cet ajout

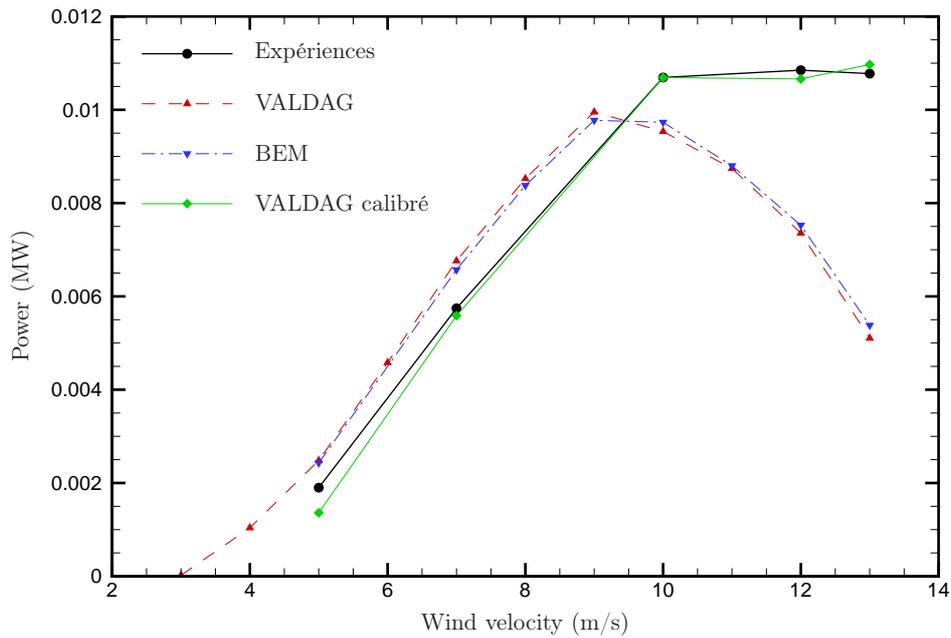


FIGURE 4.7 – Comparaison des résultats obtenus numériquement avec les résultats expérimentaux.



FIGURE 4.8 – Visitation d'une pale développée par la société Enercon.

permet de gagner 2% de puissance supplémentaire, mais reste à optimiser plus finement. La pale optimisée est représentée sur la figure 4.9. Le profil de la pale initiale reste inchangé. Elle a subi un twist moyen de 3 degrés plus un ajout d'une partie profilée proche du pied de pale. Le modèle simplifié prédit donc 7% d'augmentation de puissance. Le but sera de valider ces résultats avec le modèle détaillé.

4.2.3 Simulations tridimensionnelles et limites de l'étude

La turbine est modélisée avec les deux pales, un mat et un hub (figure 4.10). Puisque nous utilisons un maillage cartésien homogène, nous n'avons pas la prétention de modéliser et simuler (finement) les couches

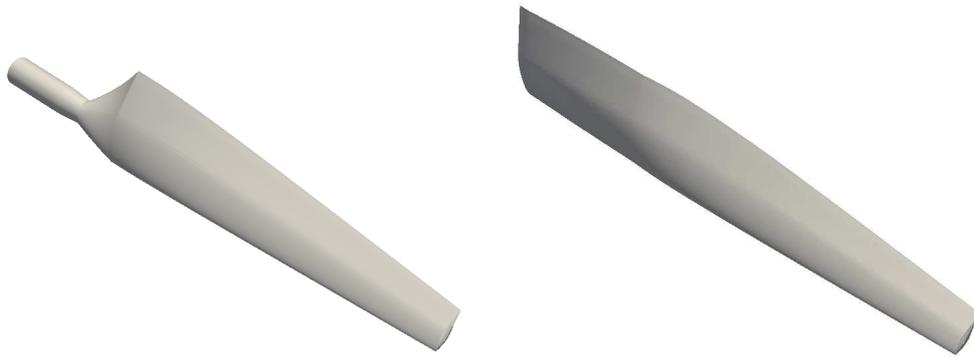


FIGURE 4.9 – Visualition de la pale initiale (NREL, à gauche), et de la pale optimisée (à droite). Le modèle simplifié prédit 7% d’augmentation de puissance.



FIGURE 4.10 – Modélisation de l’éolienne

limites. Une source possible d’erreur peut évidemment être liée à ces fines échelles qu’il est impossible de résoudre avec la présente méthode. Néanmoins, nous allons essayer d’effectuer des simulations de ce rotor. Les vitesses intervenant dans ce type de problèmes sont très élevées et nous rencontrons une turbulence pleinement développée. Il serait sans doute intéressant d’étudier finement cette turbulence, mais nous avons choisi d’utiliser un simple modèle de turbulence de type LES Smagorinsky lilly, qui, à défaut d’être très précis, permet de stabiliser les simulations numériques. A l’heure actuelle, ces simulations dites détaillées peuvent être considérées comme solution d’un modèle réduit (car toutes les échelles ne sont pas résolues).

Afin de tester et valider la méthode numérique utilisée, nous comparons les résultats obtenus avec les résultats expérimentaux publiés par le NREL et décrits au §4.2.1. A noter qu’il s’agit ici d’un cas test légèrement différent de celui traité au paragraphe précédent². Un exemple de simulation est reporté sur la figure 4.11. Le domaine est $x \in [-5, 15]$, $y \in [-12, 8]$ et $z \in [-10, 10]$. Le maillage cartésien uniforme est 400^3 , et donc les cellules ont une arête de 5cm. Il est évident dans ce cas que les couches limites ne sont

2. Ces résultats ont été effectués il y a trois ans, avant le développement du code simplifié VALDAG. Pour sa thèse Xin a choisi un cas test différent, mieux documenté.

pas calculées par la présente simulation. Des conditions aux limites de type non-réfléctives (Jin et Braza, 1993) sont imposées en sortie et des conditions de Dirichlet homogènes sont imposées sur la sol. Enfin, des conditions aux limites de types Neumann homogènes sont imposées sur les autres frontières avec également $\mathbf{u} \cdot \mathbf{n} = 0$. Sur la figure 4.11 nous voyons apparaître un sillage fortement turbulent. Par contre ce sillage

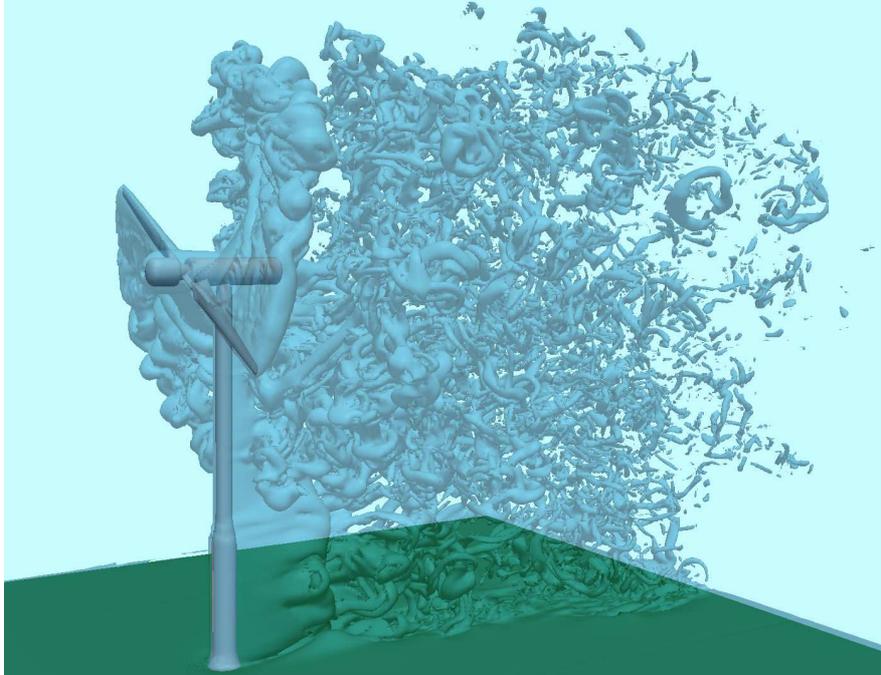


FIGURE 4.11 – Champ de vorticit e cr e par l' olienne du NREL obtenu num riquement. Vitesse du vent en entr e : 10m/s; vitesse de rotation 72 tours/minutes.

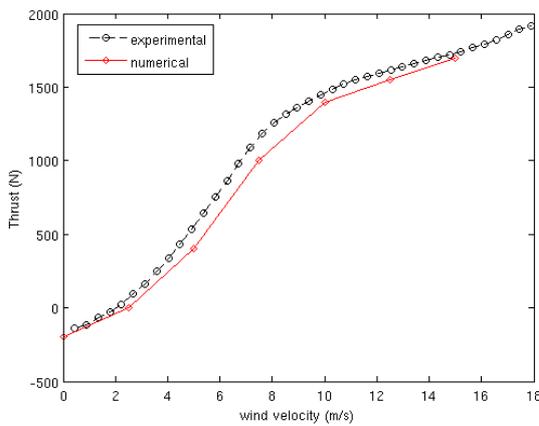
est tr s, voir trop, dissipatif. En effet, le tourbillon de bout de pale (par analogie   l'extr mit  d'une aile d'avion) dispara t tr s rapidement. Le sillage devrait avoir une forme qui ressemble au lâch  de fum e illustr  sur la figure 4.12. Bien qu'il s'agisse d'un transport d'un scalaire passif, on devine ais ment la pr sence du tourbillon. Afin d'am liorer la simulation, nous consid rons uniquement les deux pales (sans hub et mat) dans un domaine $x \in [-1.5, 2.5]$, $y \in [-8, 8]$ et $z \in [-8, 8]$. Le maillage associ  est $600 \times 400 \times 400$, et nous avons alors 125 points de maillage sur la longueur de la pale, 20 points sur la corde maximale et 20 points sur l' paisseur maximale. Ces valeurs sont en g n ral suffisantes pour capter une couche limite laminaire. En effet nous avons fait des tests de convergence pour l' coulement autour d'une sph re qui montre que 20   30 points sur le diam tre sont suffisants jusqu'  un nombre de Reynolds de 500 environ. Dans tous les cas, nous savons que ce n'est pas suffisant pour des couches limites turbulentes. Nous atteignons ici les limites de cette m thode de mod lisation et de simulation. Comme nous le verrons en conclusion de ce m moire, nous sommes en train de travailler, conjointement avec la soci t  OPTIMAD (spin off du politecnico di Torino) sur des m thodes overset de type chim re. Nous conservons un maillage de fond qui est cart sien sur lequel se superpose un maillage conforme aux pales sur lequel nous pouvons approcher un maillage de couche limite³.

Nous avons tout de m me calcul  la pouss e du rotor ainsi que la puissance extraite. Cette derni re est directement corr l e au moment suivant l'axe x et   la portance dans la direction de rotation. Les figures 4.13(a) et 4.13(b) pr sentent respectivement une confrontation entre exp rience et simulation num rique de la pouss e du rotor et de la puissance extraite pour diff rentes valeurs de vent incident avec un rotation fix e   72 tours par minutes. Les valeurs de la pouss e obtenues exp rimentalement et num riquement sont en tr s bon accord. Ceci est du principalement au fait que cette quantit  est intimement li  au ma tre couple qui est bien d termin  avec le maillage  tudi . Les valeurs de la puissance obtenues num riquement diff rent de celles obtenues exp rimentalement par un facteur d'environ deux. Les  volutions ont tout de m me une tendance similaire. Cette diff rence provient probablement du fait que la portance dans la direction normale   la rotation des pales (qui est li    la puissance) n'est pas bien calcul e. Ceci d coule du fait que la g om trie

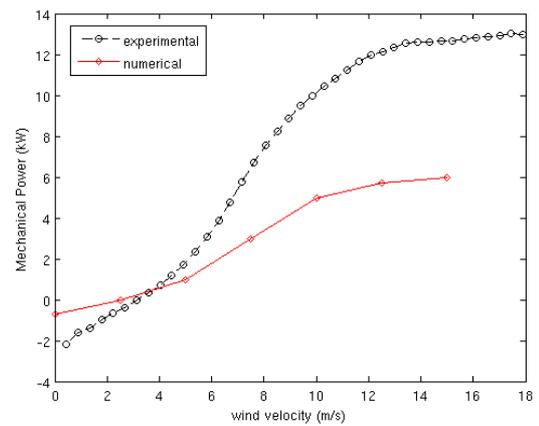
3. Raisonnablement, nous serons encore assez  loign  d'un maillage de type couche limite.



FIGURE 4.12 – Mise en évidence du tourbillon de bout de pale par un lâché de fumée.



(a) Traînée générée par les pales.



(b) Puissance extraite par les pales.

FIGURE 4.13 – Evolution de la traînée et de la puissance en fonction de la vitesse du vent.

n'est pas suffisamment bien modélisée sur le maillage étudié. On rappelle également que les couches limites turbulentes ne sont pas calculées.

A ce stade nous pouvons considérer que le modèle simplifié donne de meilleurs résultats pour les faibles vitesses de vent, mais le modèle détaillé est plus consistant sur toute la plage des vents considérée. Des tests récents montrent également une grande influence du modèle de turbulence utilisé sur nos résultats. En particulier, des résultats préliminaires montrent que "d'éteindre" continûment le modèle de turbulence proche des pales améliorent considérablement les résultats de puissance. A titre d'exemple la puissance extraite pour $V = 13 \text{ m/s}$ est de $12,5 \text{ kW}$ au lieu de 6 précédemment, contre environ 11 kW expérimentalement.

Un point qui intéresse très fortement les industriels du domaine est le point de fonctionnement optimal de l'éolienne défini par le coefficient de puissance $c_p = P / (1/2 \rho S_{ref} U_{wind}^3)$ en fonction du paramètre *tip speed ratio* $\lambda = U_{tip} / U_{wind}$, où S_{ref} est la surface (disque) de référence décrit par le mouvement des pales,

U_{wind} est la vitesse du vent incident et U_{tip} de la pale à son l'extrémité. La figure 4.14 présente les évolutions

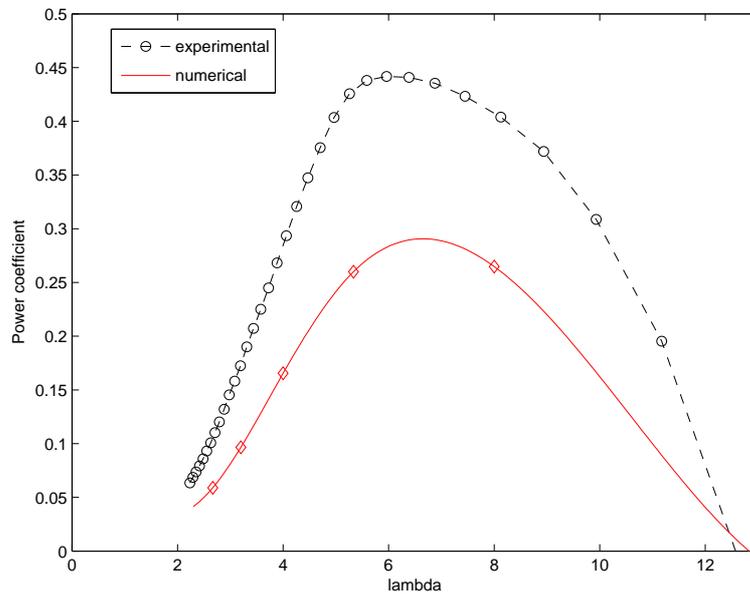


FIGURE 4.14 – Evolution du coefficient de puissance c_p en fonction du *tip speed ratio* λ .

du coefficient du puissance obtenues numériquement avec celles obtenues expérimentalement. Le maximum du coefficient de puissance est obtenu pour $\lambda = 6.5$ numériquement contre $\lambda = 5.6$ expérimentalement, soit environ 12% d'erreur. Compte tenu des limites de la méthode, ce résultat est encourageant. La suite, nous l'avons déjà mentionné, est de refaire ces simulations avec des maillages overset de type chimère, et d'affiner le modèle de turbulence.

4.3 Récupération de l'énergie de la houle

Le but de cette section est de modéliser et de simuler numériquement un dispositif capable d'extraire l'énergie de la houle. Nous avons alors pensé à des structures articulées ou élastiques. A titre d'exemple, le Pelamis est un système articulé ayant pour but de récupérer l'énergie des vagues. Ce serpent de mer géant (170 mètres) est composé de quatre cylindres (figure 4.15) reliés entre eux, et a été développé par une société écossaise. Plusieurs serpents Pelamis ont été placés au large des côtes portugaises, et de nombreux autres projets Pelamis sont en cours⁴. Ces serpents sont reliés au sol pour acheminer l'énergie capturée. Ce serpent



FIGURE 4.15 – Serpent de mer Pelamis. Crédit : Pelamis Wave Power.

articulé pourrait être généralisé avec plus de cylindres relié entre eux et allant vers un modèle élastique continu.

4. voir <http://www.pelamiswave.com>.

Afin de modéliser et simuler numériquement ce type de problème, il est nécessaire de développer plusieurs aspects :

- Développer une structure élastique articulée (§4.3.1)
- Coupler ce modèle avec un solveur fluide (§4.3.2)
- Développer des solveurs bifluides précis (§4.3.3)
- Développer des fonctions level set précises avec un modèle de mouillage de type Cox (§4.3.4)
- Coupler le modèle élastique avec le solveur bifluide et le modèle de mouillage (§4.3.5)

4.3.1 Un modèle de structure élastique

Le modèle utilisé dans la suite est inspiré par McMillen et Holmes (2006). La structure est approchée par un profil de poutre. La configuration de la structure est représentée à chaque instant t par la courbe $s \mapsto \mathbf{r}(s, t) = (x(s, t), y(s, t), z(s, t))$ qui représente la ligne médiane, que l'on appellera colonne vertébrale (figure 4.16). Nous nous limiterons à des déplacements de la colonne vertébrale dans le plan $z = 0$, et donc $s \mapsto \mathbf{r}(s, t) = (x(s, t), y(s, t))$. La longueur de la structure est ℓ , et donc $s \in [0, \ell]$. Nous imposons une contrainte d'inextensibilité, traduite mathématiquement par $|\frac{\partial \mathbf{r}}{\partial s}| = 1$. En définissant la tangente $\mathbf{t} = \frac{\partial \mathbf{r}}{\partial s} = (\cos(\phi), \sin(\phi))$ et la normale $\mathbf{n} = (-\sin(\phi), \cos(\phi))$, la condition d'inextensibilité peut se ré-écrire sous la forme

$$\frac{\partial x}{\partial s} = \cos(\phi), \quad (4.13a)$$

$$\frac{\partial y}{\partial s} = \sin(\phi). \quad (4.13b)$$

Chaque élément de la poutre est soumis à une force intérieure \mathbf{f} , un moment intérieur M , et des forces extérieures \mathbf{W} . Par définition, les forces et moments intérieurs sont exercés sur la partie $(s, s + ds)$ par la partie $[0, s)$. Dans la suite on note classiquement $\dot{a} = \frac{\partial a}{\partial t}$ et $\ddot{a} = \frac{\partial^2 a}{\partial t^2}$ pour toute variable a .

Les équations du mouvement sont alors :

$$\rho S \ddot{x} = W^x + f^x, \quad (4.14a)$$

$$\rho S \ddot{y} = W^y + f^y, \quad (4.14b)$$

$$\rho I \ddot{\phi} = \frac{\partial M^z}{\partial s} + f^y \cos(\phi) + f^x \sin(\phi), \quad (4.14c)$$

où ρ est la masse volumique de la poutre, S est la section droite et I est le moment d'inertie dans le plan (x, y) . Nous supposons que S et I dépendent de la position s . Souvent la section droite est considérée elliptique ou rectangulaire. La déformation de la poutre peut être déterminée en imposant une courbure $\kappa(s, t)$. Lorsque $\frac{\partial \phi}{\partial s} = \kappa$ la poutre est en équilibre. Nous considérons une loi constitutive linéaire classique qui pose que le moment est proportionnel à la déviation de la courbure prescrite, *i.e.*

$$M^z = EI \left(\frac{\partial \phi}{\partial s} - \kappa \right) + \mu I \frac{\partial \dot{\phi}}{\partial s}, \quad (4.15)$$

où EI est le module de torsion élastique et μI le module de torsion visqueux, avec E le module d'Young du matériau.

Le système (4.14) avec la loi constitutive (4.15) et la contrainte (4.13) peut être résolu avec des conditions initiales et aux limites. Des conditions limites naturelles pour une structure libre sont $M^z(0) = M^z(\ell) = f^y(0) = f^y(\ell) = f^x(0) = f^x(\ell) = 0$.

Discrétisation et méthode numérique

La discrétisation de la poutre est détaillée sur la figure 4.17. On considère une discrétisation de la poutre en N segments de longueurs ℓ_i , $i = 1, \dots, N$ avec $\sum_{i=0}^N \ell_i = \ell$. On note $x_i(t) = x(\sum_{j=1}^i \ell_j, t)$, de même pour les autres variables. La contrainte d'inextensibilité (4.13) devient :

$$x_{i+1} = x_i + \frac{\ell_i}{2} \cos(\phi_i) + \frac{\ell_{i+1}}{2} \cos(\phi_{i+1}) \quad (4.16a)$$

$$y_{i+1} = y_i + \frac{\ell_i}{2} \sin(\phi_i) + \frac{\ell_{i+1}}{2} \sin(\phi_{i+1}). \quad (4.16b)$$

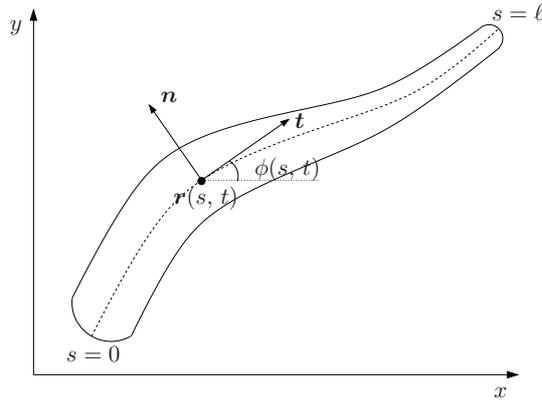
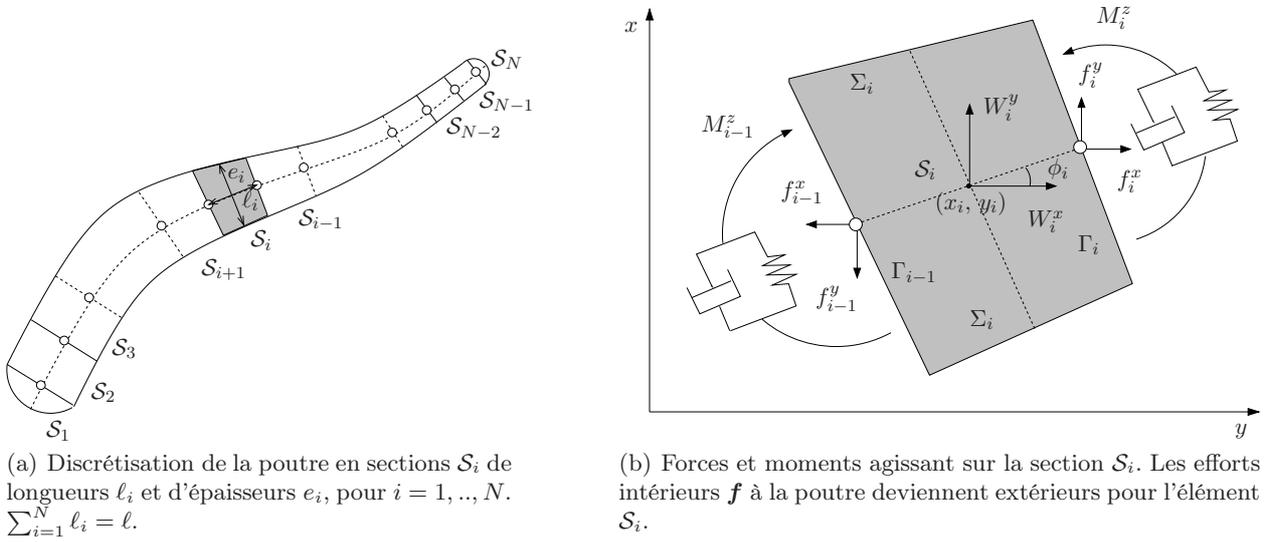
FIGURE 4.16 – Représentation de la poutre dans le plan $z = 0$.(a) Discrétisation de la poutre en sections S_i de longueurs l_i et d'épaisseurs e_i , pour $i = 1, \dots, N$. $\sum_{i=1}^N l_i = l$.(b) Forces et moments agissant sur la section S_i . Les efforts intérieurs \mathbf{f} à la poutre deviennent extérieurs pour l'élément S_i .

FIGURE 4.17 – Configuration et discrétisation de la poutre élastique.

Les équations du mouvement (4.14) sont :

$$m_i \ddot{x}_i = l_i W_i^x + f_i^x - f_{i-1}^x, \quad (4.17a)$$

$$m_i \ddot{y}_i = l_i W_i^y + f_i^y - f_{i-1}^y, \quad (4.17b)$$

$$\rho I_i l_i \ddot{\phi}_i = M_i^z - M_{i-1}^z + \frac{1}{2}(l_i f_i^y + l_{i-1} f_{i-1}^y) \cos(\phi_i) - \frac{1}{2}(l_i f_i^x + l_{i-1} f_{i-1}^x) \sin(\phi_i), \quad (4.17c)$$

où $m_i = \rho S_i l_i$ est la masse de l'élément i . La masse totale de la poutre est notée $m = \sum_{i=1}^N m_i$. Enfin, la loi constitutive devient :

$$M_i^z = E I_i \left(\frac{\phi_i - \phi_{i-1}}{l_i} - \kappa \right) + \mu I_i \left(\frac{\dot{\phi}_i - \dot{\phi}_{i-1}}{l_i} \right), \quad (4.18)$$

Le schéma numérique utilisé pour résoudre le système précédent (4.16)-(4.17)-(4.18) est décrit dans ce qui suit. La position de la poutre $(x(s, t), (y(s, t))$ est entièrement déterminée par la connaissance d'un point, par exemple $(x(0, t), y(0, t))$, et de l'angle tangent ϕ en tout point. Nous avons alors :

$$x(s, t) = x(0, t) + \int_0^s \cos(\phi(s', t)) ds', \quad (4.19a)$$

$$y(s, t) = y(0, t) + \int_0^s \sin(\phi(s', t)) ds'. \quad (4.19b)$$

D'un point de vue discret, nous avons :

$$x_i = x_1(t) + C_i(t), \quad (4.20a)$$

$$y_i = y_1(t) + S_i(t). \quad (4.20b)$$

avec

$$C_i(t) = \frac{\ell_1}{2} \cos(\phi_1) + \sum_{j=2}^{i-1} \ell_j \cos \phi_j + \frac{\ell_i}{2} \cos(\phi_i), \quad (4.21a)$$

$$S_i(t) = \frac{\ell_1}{2} \sin(\phi_1) + \sum_{j=2}^{i-1} \ell_j \sin \phi_j + \frac{\ell_i}{2} \sin(\phi_i), \quad (4.21b)$$

et $C_1 = S_1 = 0$.

Nous pouvons réécrire les équations (4.17c) et (4.17b) en utilisant (4.20) pour $i = 1, \dots, N$:

$$\ddot{x}_1 = -\ddot{C}_i + (\ell_i W_i^x + f_i^x - f_{i-1}^x) / m_i, \quad (4.22a)$$

$$\ddot{y}_1 = -\ddot{S}_i + (\ell_i W_i^y + f_i^y - f_{i-1}^y) / m_i. \quad (4.22b)$$

Il en découle, pour $i = 1, \dots, N - 1$:

$$-\ddot{C}_i + (\ell_i W_i^x + f_i^x - f_{i-1}^x) / m_i = -\ddot{C}_{i+1} + (\ell_{i+1} W_{i+1}^x + f_{i+1}^x - f_i^x) / m_{i+1}, \quad (4.23a)$$

$$-\ddot{S}_i + (\ell_i W_i^y + f_i^y - f_{i-1}^y) / m_i = -\ddot{S}_{i+1} + (\ell_{i+1} W_{i+1}^y + f_{i+1}^y - f_i^y) / m_{i+1}. \quad (4.23b)$$

Nous voyons apparaître le fait que f_i^x dépend de f_{i-1}^x et f_{i+1}^x (de même pour f^y). En posant $f_0^x = f_N^x = f_0^y = f_N^y = 0$, on obtient les systèmes linéaires $N - 1 \times N - 1$ suivants :

$$\tilde{A} \tilde{\mathbf{f}}^x = \mathbf{b}^x \quad \text{et} \quad \tilde{A} \tilde{\mathbf{f}}^y = \mathbf{b}^y, \quad (4.24)$$

avec :

$$\tilde{\mathbf{f}}^x = (f_1^x, f_2^x, \dots, f_{N-1}^x)^T, \quad (4.25a)$$

$$\tilde{\mathbf{f}}^y = (f_1^y, f_2^y, \dots, f_{N-1}^y)^T, \quad (4.25b)$$

$$\mathbf{b}_i^x = \frac{d^2}{dt^2} \left(\frac{\ell_i}{2} \cos(\phi_i) + \frac{\ell_{i+1}}{2} \cos(\phi_{i+1}) \right) + \frac{\ell_i}{m_i} W_i^x + \frac{\ell_{i+1}}{m_{i+1}} W_{i+1}^x, \quad i = 1, \dots, N - 1, \quad (4.25c)$$

$$\mathbf{b}_i^y = \frac{d^2}{dt^2} \left(\frac{\ell_i}{2} \sin(\phi_i) + \frac{\ell_{i+1}}{2} \sin(\phi_{i+1}) \right) + \frac{\ell_i}{m_i} W_i^y + \frac{\ell_{i+1}}{m_{i+1}} W_{i+1}^y, \quad i = 1, \dots, N - 1, \quad (4.25d)$$

$$\tilde{A} = \begin{pmatrix} -\left(\frac{1}{m_1} + \frac{1}{m_2}\right) & \frac{1}{m_2} & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{m_2} & -\left(\frac{1}{m_2} + \frac{1}{m_3}\right) & \frac{1}{m_3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{m_{N-1}} & -\left(\frac{1}{m_{N-1}} + \frac{1}{m_N}\right) \end{pmatrix}$$

Les systèmes linéaires (4.24) peuvent facilement être résolus pour obtenir $\tilde{\mathbf{f}}^x$ et $\tilde{\mathbf{f}}^y$. Nous verrons par la suite que ces systèmes ne sont jamais résolus, mais que les efforts \mathbf{f}^x et \mathbf{f}^y jouent uniquement le rôle de multiplicateurs de Lagrange pour imposer les contraintes.

Notons A la matrice $N \times N$ définie par :

$$A = \begin{pmatrix} \tilde{A}^{-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

En supposant que les forces de contact sont nulles aux extrémités, $f_0^x = f_N^x = f_0^y = f_N^y = 0$, nous avons :

$$\mathbf{f}^x = A (-H(\cos(\phi)) \dot{\phi}^2 - H(\sin(\phi)) \ddot{\phi} + \tilde{\mathbf{W}}^x), \quad (4.26a)$$

$$\mathbf{f}^y = A (-H(\sin(\phi)) \dot{\phi}^2 - H(\cos(\phi)) \ddot{\phi} + \tilde{\mathbf{W}}^y), \quad (4.26b)$$

avec

$$\widetilde{W}^x = \begin{pmatrix} \frac{\ell_1}{m_1} W_1^x - \frac{\ell_2}{m_2} W_2^x \\ \vdots \\ \frac{\ell_{N-1}}{m_{N-1}} W_{N-1}^x - \frac{\ell_N}{m_N} W_N^x \\ 0 \end{pmatrix}, \quad (4.27a)$$

$$\widetilde{W}^y = \begin{pmatrix} \frac{\ell_1}{m_1} W_1^y - \frac{\ell_2}{m_2} W_2^y \\ \vdots \\ \frac{\ell_{N-1}}{m_{N-1}} W_{N-1}^y - \frac{\ell_N}{m_N} W_N^y \\ 0 \end{pmatrix}, \quad (4.27b)$$

et

$$H(\mathbf{z}) = \frac{1}{2} \begin{pmatrix} \ell_1 z_1 & \ell_2 z_2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \ell_2 z_2 & \ell_3 z_3 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \ell_{N-1} z_{N-1} & \ell_N z_N \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Finalement, l'équation des moments (4.18) s'écrit :

$$\begin{aligned} & (J - G(\cos(\phi)) A H(\cos(\phi)) - G(\sin(\phi)) A H(\sin(\phi))) \ddot{\phi} \\ & = (-G(\cos(\phi)) A H(\sin(\phi)) + G(\sin(\phi)) A H(\cos(\phi))) \dot{\phi} \\ & \quad + \widetilde{M}^z + G(\cos(\phi)) A \widetilde{W}^y - G(\sin(\phi)) A \widetilde{W}^x. \end{aligned} \quad (4.28)$$

où J est la matrice d'inertie avec $J_i = \rho I_i \ell_i$, et

$$G(\mathbf{z}) = \frac{1}{2} \begin{pmatrix} \ell_1 z_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \ell_2 z_2 & \ell_2 z_2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \ell_3 z_3 & \ell_3 z_3 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \ell_{N-1} z_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \cdots & \ell_N z_N & \ell_N z_N \end{pmatrix},$$

et

$$\widetilde{M}^z = \begin{pmatrix} M_1^z \\ M_2^z - M_1^z \\ \vdots \\ M_{N_1}^z - M_{N_2}^z \\ -M_{N_1}^z \end{pmatrix}.$$

En sommant les équations (4.22) de $i = 1, \dots, N$, on obtient :

$$\ddot{x}_1 = \frac{1}{m} \sum_{i=1}^N (\ell_i W_i^x - m_i \ddot{C}_i), \quad (4.29a)$$

$$\ddot{y}_1 = \frac{1}{m} \sum_{i=1}^N (\ell_i W_i^y - m_i \ddot{S}_i). \quad (4.29b)$$

Notons que dans le cas d'une structure pesante sous l'effet de la gravité $\mathbf{g} = g \mathbf{e}_y$, nous avons :

$$\ddot{x}_1 = \frac{1}{m} \sum_{i=1}^N (\ell_i W_i^x - m_i \ddot{C}_i), \quad (4.30a)$$

$$\ddot{y}_1 = \frac{1}{m} \sum_{i=1}^N (\ell_i W_i^y - m_i \ddot{S}_i - m_i g). \quad (4.30b)$$

Résumé : finalement, la résolution de l'équation (4.28) nous donne ϕ et la résolution des équations (4.29) (ou (4.29)) nous donne x_1 et y_1 . La connaissance du vecteur $\mathbf{z} = (x_1, y_1, \phi_1, \dots, \phi_N)^T$ nous donne la position de la poutre en tout point. Le système à résoudre s'écrit $\ddot{\mathbf{z}} = G(\mathbf{z}, \dot{\mathbf{z}}, t)$, et peut être résolu efficacement à l'aide d'une méthode de Runge-kutta. Les pas de temps de discrétisation dépendent des modules d'Young et du coefficient d'amortissement, mais sont généralement très petits, de l'ordre de 10^{-6} . La résolution de l'équation (4.28) est relativement coûteuse car nous devons inverser des opérateurs et résoudre des systèmes linéaires de taille environ $N \times N$ avec généralement $N = 100$ éléments pour la poutre. Ces petits pas de temps couplés aux coûts numériques par itération rendent la résolution de la structure relativement coûteux d'un point de vue temps CPU.

4.3.2 Résolution numérique et couplage

Jusqu'à présent nous avons imposé une action de déformation de la structure engendrant des efforts sur le fluide. Cette action amenait une réaction opposée qui faisait avancer la structure. Il s'agissait d'une interaction unidirectionnelle : la déformation de la structure pouvait se faire sans considération des possibles efforts du fluide. Dans le cas d'une structure élastique, ce n'est plus du tout le cas. Il n'est plus possible d'imposer exactement la déformation sans considérations de l'environnement extérieur. Le principe d'action réaction doit être satisfait, ou du moins le moins violé d'un point de vue numérique. La violation de ce principe amène en général un surplus d'énergie qu'il faudrait dissiper. Des techniques de dissipation visqueuses pourraient être mises en place localement, mais modifieraient le comportement du système. Il est alors nécessaire de bien calculer la réaction liée à l'action (de déformation par exemple), et de façon similaire de bien prédire une action admissible pour cette réaction. On voit apparaître la notion de couplage fort, qui d'un point de vue numérique représente un couplage en temps implicite entre le fluide et la structure. Le couplage en espace pouvant être réalisé avec une des méthodes présentées au chapitre 2, à savoir pénalisation, frontières immergées ou IPC.

Nous pouvons développer des méthodes très précises et stables en temps pour le fluide d'un côté et pour la structure de l'autre. Par contre il est bien plus délicat de développer des méthodes stables pour le couplage fluide/structure. Ce phénomène est d'autant plus perceptible lorsque les densités du fluide et de la structure sont proches et où l'effet de masse ajoutée est présent.

Trois classes d'algorithmes de couplage en temps sont généralement utilisés :

- Algorithme décalé : chaque sous-système est avancé en temps successivement,
- Algorithme parallèle : les deux sous-système sont avancés simultanément,
- Algorithme itéré : chaque pas de temps de la simulation peuvent engendrer des pas de temps fictifs pour faire converger une certaine quantité afin de minimiser l'énergie artificielle qui pourrait être créée.

Dans chaque classe d'algorithme, nous pouvons choisir différentes informations à échanger entre le fluide et la structure : efforts, position de l'interface, etc ..

Pour les premiers exemples de nage nous avons utilisé un algorithme de type couplage décalé. L'algorithme parallèle est très complexe pour des cas non monolithiques comme dans notre cas. En effet les pas de temps liés à la structure sont beaucoup plus petits que ceux liés au fluide. A titre d'exemple, les pas de temps généralement utilisés pour le fluide sont de l'ordre de $10^{-2} - 10^{-3}$ et ceux pour la structure de l'ordre de 10^{-6} . Les pas de temps pour la structure dépendent de module d'Young et du coefficient d'amortissement visqueux. Plus ces paramètres sont grands et plus le pas de temps est petit.

Nous préférons utiliser un algorithme itéré. A un pas de temps donné, nous souhaitons déformer la structure avec les efforts du fluide sur la structure \mathbf{W}^{n+1} (implicite) et non \mathbf{W}^n (explicite, algorithme décalé). Ce processus est donc effectué par une méthode itérative. Un algorithme simple est le suivant :

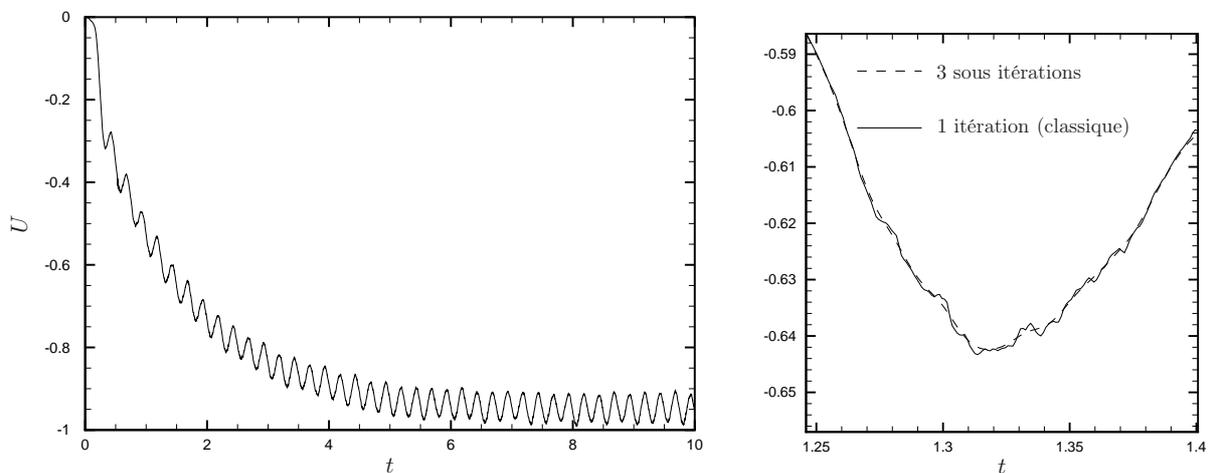
1. Au pas de temps t^n l'interface est capturée par la level set ϕ^n . A cet instant on impose éventuellement une courbure κ^n . On pose l'itération fictive $k = 0$ et les efforts du fluide sur la structure sont $\widetilde{\mathbf{W}}_k^n = \mathbf{W}^n$.
2. Code structure : à partir de ϕ^n calculer $\widetilde{\phi}_k^{n+1}$ avec κ^n et $\widetilde{\mathbf{W}}_k^n$,
3. Code fluide : à partir du déplacement prédit entre ϕ^n et $\widetilde{\phi}_k^{n+1}$, calculer les efforts $\widetilde{\mathbf{W}}_{k+1}^n$,
4. si $e = \|\widetilde{\mathbf{W}}_{k+1}^n - \widetilde{\mathbf{W}}_k^n\|_\infty < \epsilon$, alors on pose $n = n + 1$ et on retourne à l'étape 1. Sinon, on pose $k = k + 1$ et on retourne à l'étape 2.

Comme nous venons de le préciser, les pas de temps de la structure sont bien plus petits que les pas de temps du fluide. A l'étape 2 de l'algorithme précédant, des sous itérations doivent être réalisés pour la structure.

En général, 3 – 4 sous itérations k sont suffisantes pour obtenir une convergence satisfaisante, parfois moins pour des larges ratios de densités. Néanmoins, cet algorithme reste très coûteux numériquement.

Exemple d'extraction de l'énergie des tourbillons : nage d'un poisson inerte

Tout d'abord, nous allons valider la méthode qui couple solveur fluide avec le solveur de la structure élastique. Nous allons étudier brièvement la nage d'un poisson 2D comme il a été fait dans le chapitre 3. Nous imposons la déformation de la ligne médiane (colonne vertébrale) avec la loi de nage sinusoïdale introduite au chapitre précédent. Plus précisément, nous imposons une courbure à la colonne vertébrale. Plus le module d'Young est élevé, et plus le corps obéira à la déformation (à μ constant). Nous avons choisi $E = 10MPa$ et $\mu = 100Pa.s$. Nous avons choisi par ailleurs les mêmes paramètres que pour le poisson thunniforme introduit à la section §3.3.3. La figure 4.18 présente l'évolution temporelle de la vitesse de nage du poisson. Cette figure peut être comparée à la figure 3.7. Nous voyons une très bonne concordance de la vitesse obtenue avec ce modèle élastique et celle obtenue avec un poisson non élastique. La seule différence est que l'accélération pour atteindre la vitesse limite semble être un peu moins élevée avec le modèle élastique. Notons également que 3 sous itérations sont suffisantes pour réduire l'instabilité⁵ numérique (figure 3.7).



(a) Avec 3 sous itérations.

(b) Zoom et comparaison en fonction du nombre de sous itérations.

FIGURE 4.18 – Evolution temporelle de la vitesse du poisson avec loi de nage imposée.

Nous allons maintenant étudier un premier système de récupération de l'énergie. En effet, Beal *et al.* (2006) ont montré qu'un poisson mort (anesthésié en fait) est capable de "nager" par couplage de son corps flexible avec les tourbillons. De façon générale il a été démontré qu'un poisson est capable de diminuer son activité musculaire pour nager en utilisant les tourbillons présents dans l'écoulement (Liao *et al.*, 2003). Nous nous intéressons ici à la nage passive du "poisson mort". La configuration d'écoulement est la suivante. Nous considérons l'écoulement de sillage généré par un demi cylindre circulaire de diamètre 5 cm avec une vitesse amont $V = 57$ cm/s. Un poisson mort de longueur 17.8 cm est placé dans le sillage à une distance $D = 0.2$ du cylindre. Le nombre de Reynolds basé sur le diamètre du cylindre est d'environ 28 000. La zone d'aspiration (gradient de pression négatif) s'étend jusqu'à environ $1.75 D$ derrière le demi cylindre. La configuration est donnée sur la figure 4.19. Nous voyons sur cette figure que le poisson mort est capable de nager pour remonter le courant. Ceci peut nous faire penser également à la truite ou au saumon qui profite de la présence de rochers pour contrer le courant des rivières.

A l'équilibre, la ligne moyenne du poisson est supposée être à courbure nulle, et nous posons alors $\kappa = 0$ dans la loi constitutive (4.15). Outre le fait que ce système peut être numériquement instable (des sous itérations sont nécessaires), la nature physique du système peut également rendre ce système instable.

5. Il s'agit ici que de petites oscillations numériques. Dans certains cas ces oscillations peuvent amener le système à diverger.

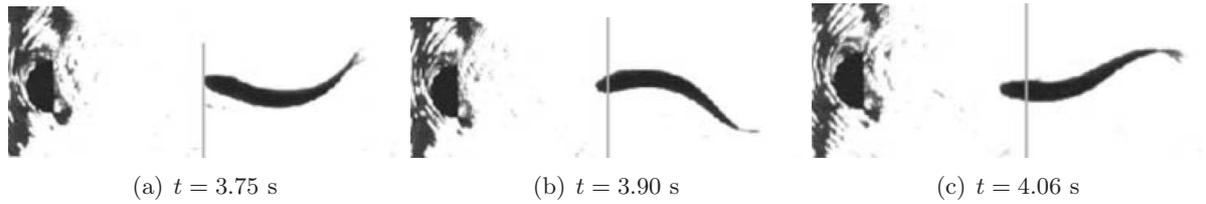


FIGURE 4.19 – Nage passive d'un poisson d'après Beal *et al.* (2006).

Dans tous les cas testés avec les paramètres mentionnés précédemment, nous avons trouvé la configuration instable physiquement. Par exemple, comme le montre la figure 4.20, pour $\mu_f = 10^{-6}$ et $E = 1MPa$ et $\mu_s = 0,01Pa/s$, le poisson n'est pas en mesure de profiter du sillage pour remonter l'écoulement. Il est rapidement entraîné dans le sillage. Nous avons alors choisi une viscosité du fluide dix fois plus grande, à

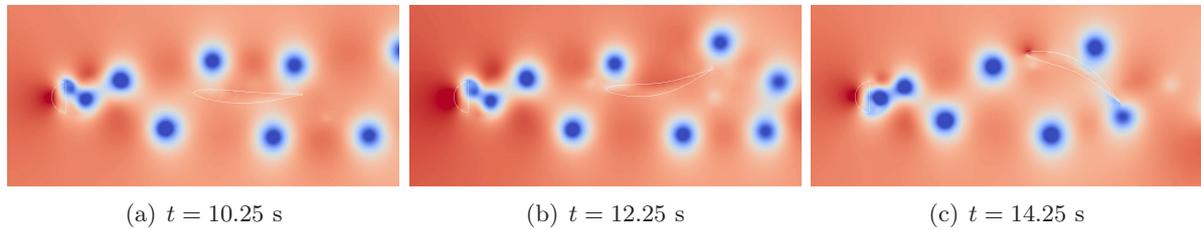


FIGURE 4.20 – Poisson élastique dans un sillage (champ de pression) pour $\mu_f = 10^{-6}$ et $E = 1MPa$ et $\mu_s = 0,01Pa/s$.

savoir $\mu_f = 10^{-5}$. La figure 4.21 présente le transport du poisson dans le sillage pour $E = 0.1MPa$ et $\mu_s = 0,1Pa/s$. Là encore, la configuration est instable et le poisson est emporté dans le sillage. Finalement,

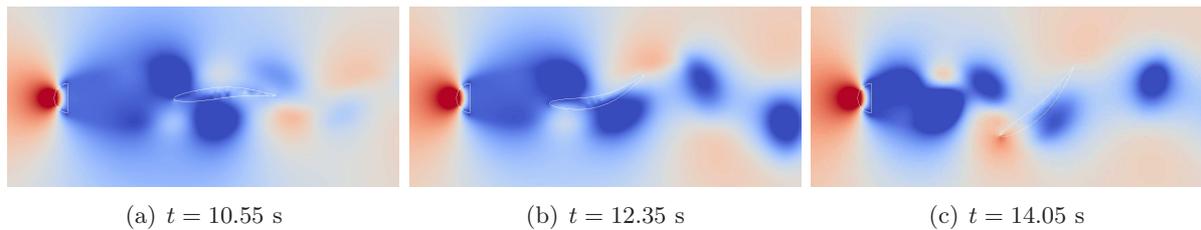


FIGURE 4.21 – Poisson élastique dans un sillage (champ de pression) pour $\mu_f = 10^{-5}$ et $E = 0.1MPa$ et $\mu_s = 0,1Pa/s$.

la figure 4.21 présente une configuration où le poisson est capable de remonter le sillage, pour $E = 1MPa$ et $\mu_s = 0,1Pa/s$. Le poisson semble nager et avancer vers le demi cylindre. Il faut néanmoins préciser que le poisson ne nage pas réellement, il profite simplement d'un gradient de pression favorable pour avancer. La configuration est toutefois stable. Les paramètres d'élasticité du poisson semblent jouer un rôle très important. En effet la structure doit vraisemblablement rentrer en résonance avec les tourbillons pour profiter de l'énergie du sillage. Une étude plus approfondie est en cours.

4.3.3 Modélisation et simulations numériques d'écoulements bifluïdes

Les écoulements bifluïdes ont été modélisés et simulés numériquement en utilisant la méthode CSF présentée au chapitre 2. L'interface est définie par une fonction level set distance signée. L'évolution de l'interface est déterminée par résolution d'une équation de transport (RK3 TVD+WENO5) puis redistanciée (à l'ordre

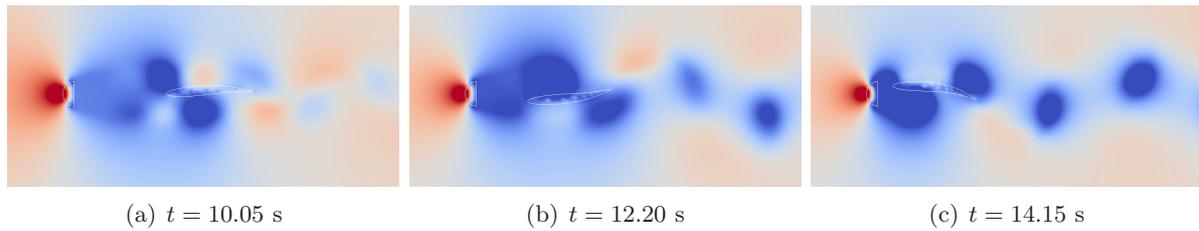


FIGURE 4.22 – Poisson élastique dans un sillage (champ de pression) pour $\mu_f = 10^{-5}$ et $E = 1MPa$ et $\mu_s = 0,1Pa/s$.

3) quand celle ci s'éloigne trop de la fonction distance. Cette partie sur les level set est effectuée et validée par le travail de post doc de Franky Luddens (post doc inria). Nous travaillons actuellement avec Lisl Weynans sur la résolution du problème elliptique (à l'ordre deux) avec une méthode de type GFM (plus précis que la méthode CSF). Cette partie a été initiée récemment et est en cours de développement. Nous présentons ici les premières validations et premiers résultats obtenus. Dans toute la section 4.3.3 nous éviterons les interactions fluide/fluide/solide. Nous choisirons alors des conditions aux limites appropriées.

Instabilité de Rayleigh-Taylor

Cette sous section a pour but de présenter une première application d'un écoulement bifluide : l'instabilité de Rayleigh-Taylor. Nous n'avons pas pour prétention d'étudier en détail cette instabilité, mais simplement de voir si les principales caractéristiques sont présentes. Pour plus de détails sur cette instabilité le lecteur curieux pour se référer à Kull (1991).

L'instabilité de Rayleigh-Taylor apparaît lorsqu'une couche de fluide densité ρ_1 est placée au dessus d'un autre fluide de densité $\rho_0 < \rho_1$. Afin de favoriser l'apparition de cette instabilité nous perturbons l'interface entre les deux fluides comme présenté sur la figure 4.3.3. Le domaine est un rectangle avec des conditions de Dirichlet homogènes pour les composantes de vitesse u et v sur toutes les frontières, exceptée des conditions de Neumann homogènes sur les bords latéraux pour v . De cette façon, nous n'avons pas d'interaction fluide/fluide/structure qui sera l'objet de la sous section suivante.

Le développement de l'instabilité est simulé et représenté sur la figure 4.23. Nous voyons apparaître un développement typique en forme de "champignon".

Goutte 3D

Nous simulons ici la chute d'une goutte d'eau en trois dimensions. Les masse volumiques sont $\rho_{air} = 1.2kg/m^3$ et $\rho_{eau} = 1000kg/m^3$, les viscosités dynamiques sont $\mu_{air} = 18.7 \cdot 10^{-6}kg/(m.s)$ et $\mu_{eau} = 10^{-3}kg/(m.s)$. Le coefficient de tension de surface entre air et eau est $\gamma = 75 \cdot 10^{-3}N/m$. Le diamètre initial de la "goutte" d'eau de 1 cm. La figure 4.24 représente l'interface air/eau à différents instants. L'interface est capturée par une fonction level set calculée par un schéma de type HJ-WENO à l'ordre 5 et est redistanciée avec la méthode introduite par Chéné *et al.* (2007). La perte de volume est de moins de 1% entre $t = 0$ et $t = 0.150$.

4.3.4 Gestion de la ligne triple

Nous avons fait le choix d'utiliser un modèle de Cox. La vitesse de l'interface fluide/fluide le long de la paroi solide est donnée en fonction de la différence entre l'angle de contact dynamique θ_D et l'angle de contact à l'équilibre θ_E . Différentes lois ont été testées et nous avons finalement retenu la loi présentée par Pinilla Velandia (2012). Notons enfin, que même en absence de modèle de gestion de la ligne triple, cette dernière se déplace sous l'effet de la dissipation induite par les schémas numériques utilisés. Ce déplacement est toutefois moins réaliste que celui observé avec le modèle de Cox.

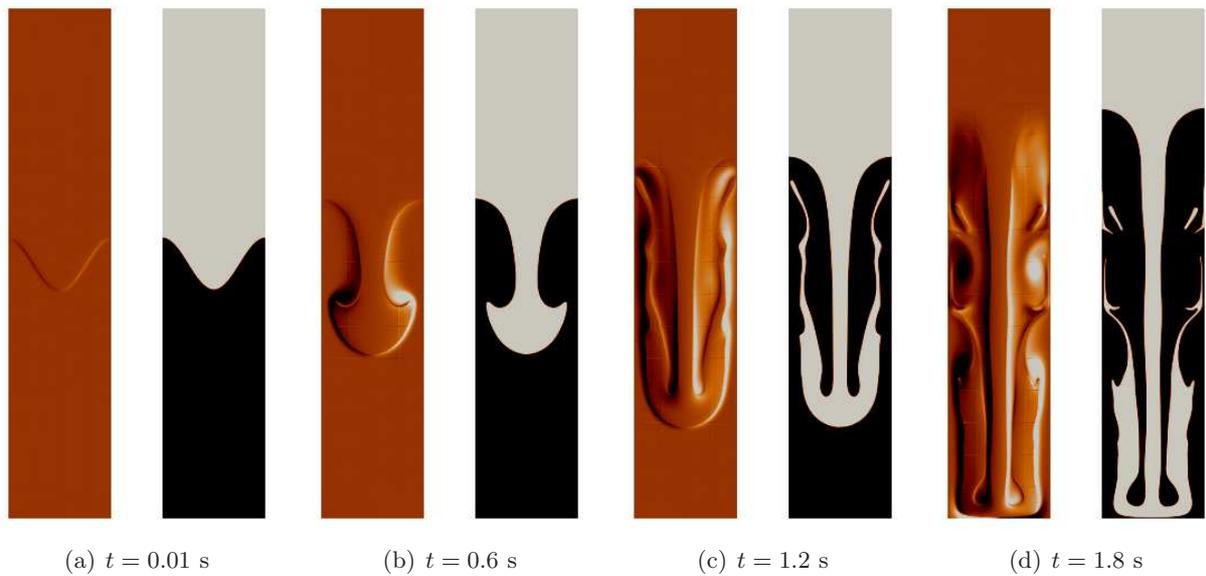


FIGURE 4.23 – Simulation num rique de l'instabilit  de Rayleigh-Taylor (vorticit    gauche et densit    droite). $\rho_{haut} = 3$, $\rho_{bas} = 1$.

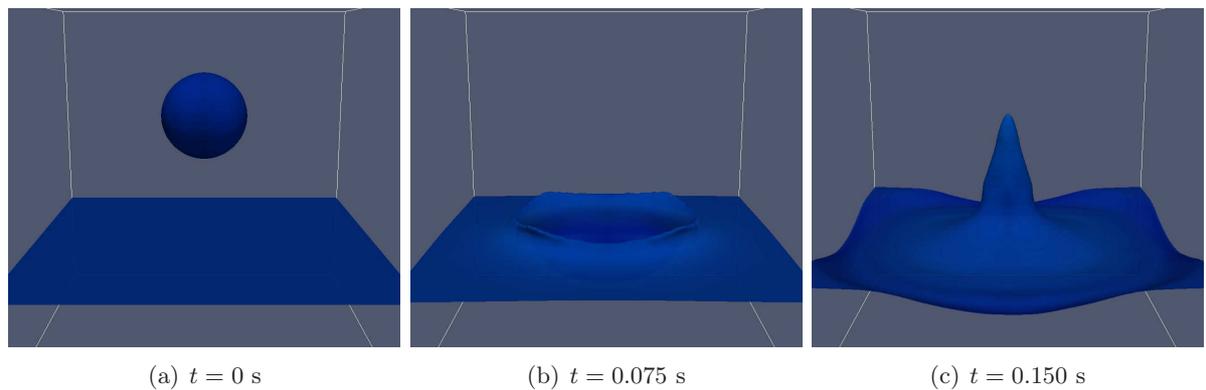


FIGURE 4.24 – Simulation de la chute d'une goutte d'eau.

Un exemple d'interface fluide/fluide/solide : navigation d'un bateau

Nous nous int ressons ici   l'exemple le plus naturel pour un  coulement pr sentant des interfaces fluide/fluide/solide : la navigation d'un bateau sur l'eau. Les masses volumiques sont $\rho_{air} = 1.2kg/m^3$ et $\rho_{eau} = 1000kg/m^3$, les viscosit s dynamiques sont $\mu_{air} = 18.7 \cdot 10^{-6}kg/(m.s)$ et $\mu_{eau} = 10^{-3}kg/(m.s)$. Le coefficient de tension de surface entre air et eau est $\gamma = 75 \cdot 10^{-3}N/m$. Le bateau a une longueur de 1 m avec une vitesse  gale   2 m/s . Le domaine  tudi  est $D = [-4.2, 1.3] \times [-0.5, 0.5] \times [-1, 1]$ avec $704 \times 128 \times 256 \approx 25 \cdot 10^5$ n uds. Nous avons utilis  la m thode CSF pr sent e au chapitre 2, o  l'interface est liss e sur 4 mailles. Finalement, afin d' viter les trop forts nombres de Reynolds, nous avons utilis  des viscosit s dix plus grandes que celles mentionn es ci-dessus.

La figure 4.26 pr sente la d formation de la surface (interface air/eau) apr s le passage du bateau. Nous pouvons observer un sillage de vagues typique de celui g n r  par un bateau. Notons que le r le de la tension de surface est tr s limit    ces  chelles en espace de l'ordre du m tre.

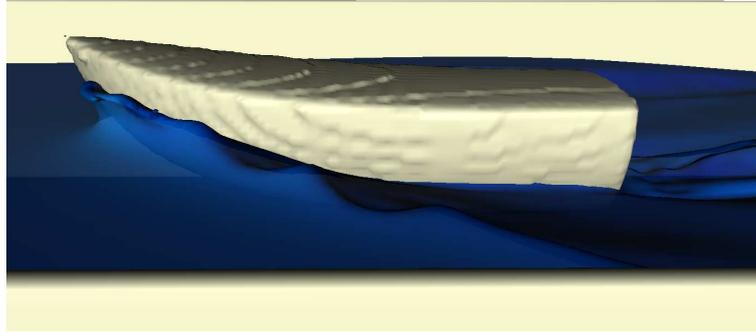


FIGURE 4.25 – Exemple de simulation d'un bateau. La longueur du bateau est $2m$ et sa vitesse est $2m/s$. La ligne triple se déplace sur la coque.

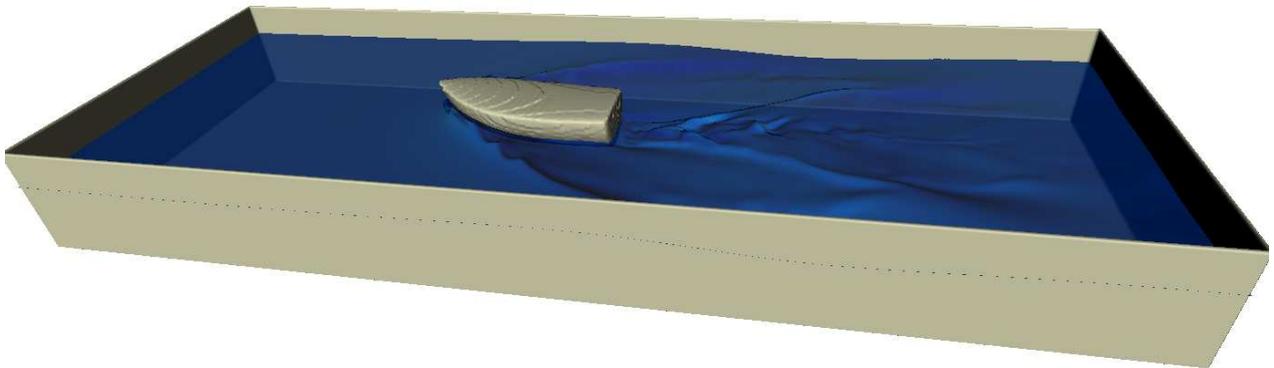


FIGURE 4.26 – Sillage généré par un bateau. La longueur du bateau est $1m$ et sa vitesse est $2m/s$.

4.3.5 Vers les serpents de mer

Nous allons maintenant nous intéresser à la modélisation et simulation de serpents de mer. La simulation de tels objets est très délicate, faisant intervenir plusieurs ingrédients. Il est nécessaire de simuler une interaction dite *two way* entre une structure élastique et deux fluides de façon implicite (ou semi implicite), et de calculer l'avancement de la ligne triple. De part la complexité de cette étude et des temps de calculs très importants engendrés par la méthode de couplage entre le solveur fluide et le solveur pour la structure élastique, nous nous limiterons pour le moment à des configurations en deux dimensions. Le développement en trois dimensions est en cours. Nous présentons ici les résultats préliminaires obtenus. La structure élastique de longueur $\ell = 1m$ est composée de 20 charnières élastiques réparties uniformément. Les masses volumiques sont $\rho_{air} = 1.2kg/m^3$ et $\rho_{eau} = 1000kg/m^3$, les viscosités dynamiques sont $\mu_{air} = 18.7 \cdot 10^{-6}kg/(m.s)$ et $\mu_{eau} = 10^{-3}kg/(m.s)$. La masse volumique de la structure est $\rho_s = 600kg/m^3$. Comme précédemment, l'interface est lissée sur 4 mailles et nous avons utilisé des viscosités dix plus grandes que celles mentionnées ci-dessus. La figure 4.27 présente l'évolution du serpent de mer ($E = 10MPa$ et $\mu = 100Pa.s$) au passage d'une vague artificielle. Cette vague est initialisé sur la figure 4.27(a). Cette vague "déferle" ensuite sur les figures 4.27(b) à 4.27(d). Le serpent de mer s'adapte parfaitement à la surface définie par l'interface air/eau. Même si la méthode générale qui couple solveur bifluide et structure élastique reste à valider, ces résultats sont tout à fait cohérents par rapport à ce que l'on s'attend à obtenir. La structure élastique flotte et se déforme de façon cohérente. Nous avons testé des structures plus au moins raides, et les résultats sont également cohérents. En particulier, la figure 4.28 représente la déformation d'une structure moins rigide ($E = 3MPa$ et $\mu = 100Pa.s$) au passage de la vague. Nous voyons que la structure élastique se déforme beaucoup et s'adapte bien à l'interface eau/air.

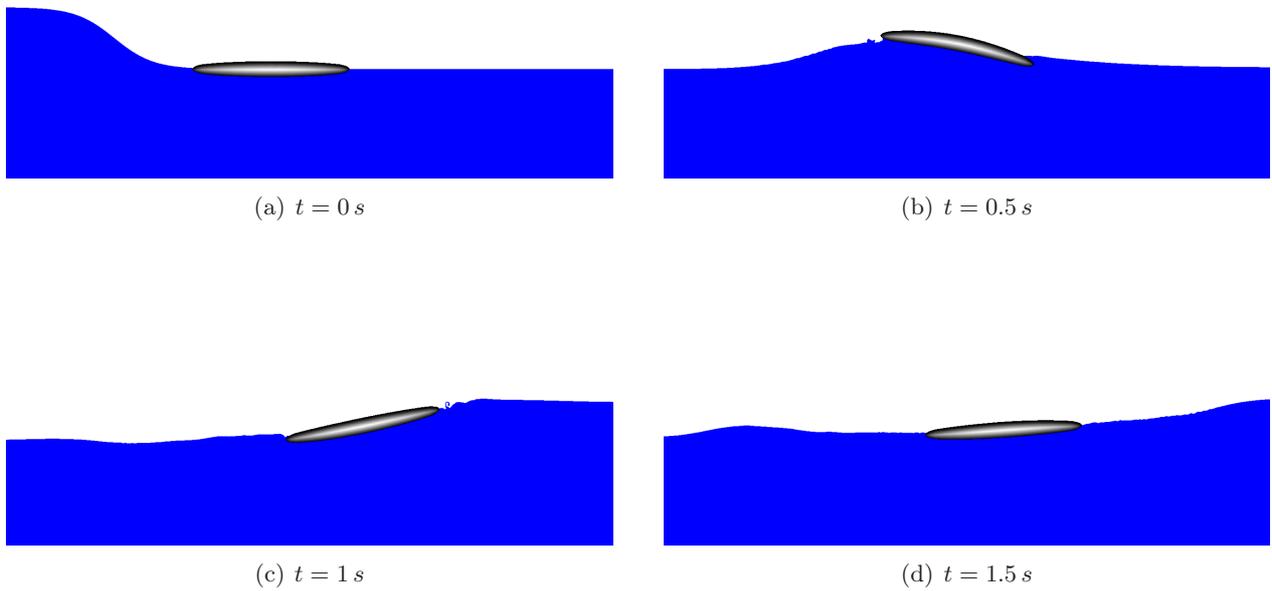


FIGURE 4.27 – Déformation de la structure élastique "rigide" sous l'effet d'une vague déferlante.

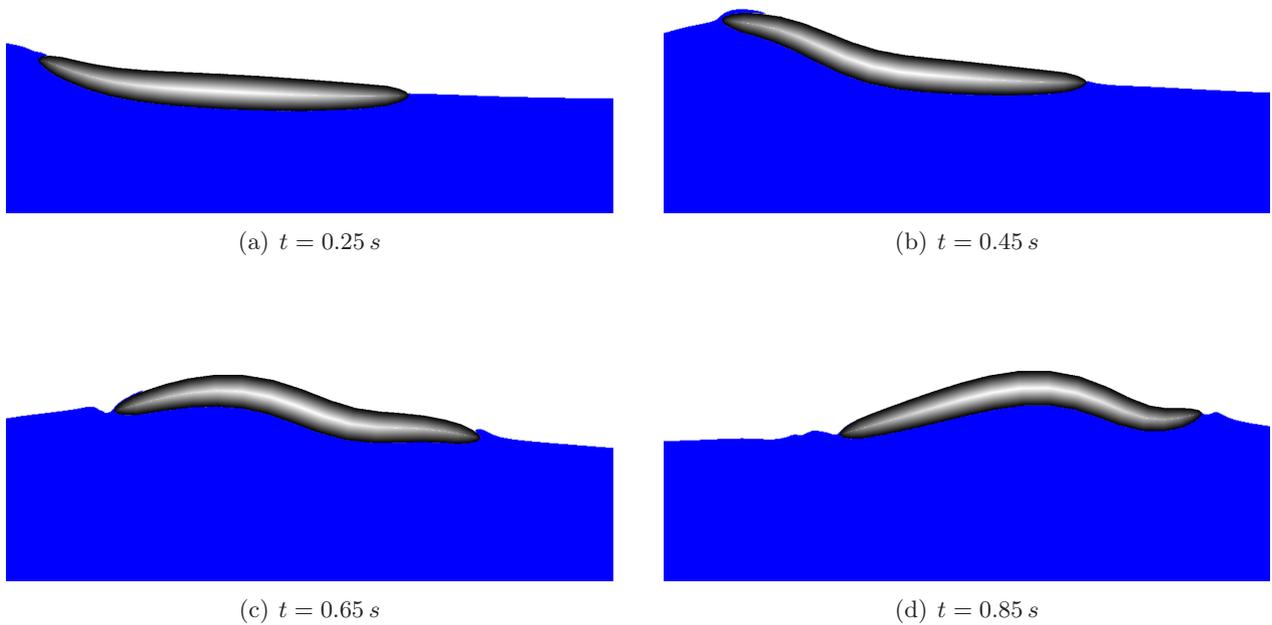


FIGURE 4.28 – Déformation de la structure élastique "molle" sous l'effet d'une vague déferlante.

4.4 Conclusions et perspectives

Ce chapitre prospectif avait pour but de présenter notre travail effectué autour des systèmes de récupération de l'énergie. Deux applications ont été étudiées.

La première est l'étude des éoliennes, avec en particulier l'objectif d'étudier l'apport énergétique d'ajout d'appendices à des pales existantes. Ce travail est réalisé dans le cadre d'une collaboration industrielle avec le groupe VALOREM et sa filiale VALEOL localisée à Bègles. Nous avons développé un modèle simplifié relativement précis qui nous a permis d'optimiser l'angle d'attaque de la pale ainsi que d'ajouter une section profilée proche du pied de pale. Le modèle simplifié prédit un gain de puissance égal à 7%. Cette pale doit être validée avec le modèle détaillé de Navier-Stokes en trois dimensions avec le code NaSCar. Pour le moment le code ne permet d'obtenir des simulations fiables. En effet, les couches limites turbulentes ne sont pas calculées. Les résultats sont néanmoins relativement satisfaisants. Un travail sur ces couches limites doit être effectué. Nous envisageons de développer des méthodes overset de type chimère avec OPTIMAD engineering (spin off de politecnico di Torino). Ces méthodes devraient permettre, pour des coûts de calculs "raisonnables", de simuler les couches limites, ou du moins d'en simuler une approximation acceptable.

La seconde application est centrée autour de systèmes articulés posés sur la surface de l'océan pour extraire l'énergie de la houle. Ces systèmes articulés sont couramment appelés serpents de mers. Ces structures créent de l'énergie en se déformant à la surface. Afin de simuler ce type de problèmes, nous avons développé un modèle de structure élastique assimilée à une poutre. Ce modèle de structure a été couplé au code fluide à l'aide d'un algorithme "implicite" itéré. Nous avons validé le solveur bifluide (air/eau), puis le couplage avec une structure solide. La ligne triple est modélisée avec un modèle simplifié de type Cox. Nous avons ensuite simulé la nage d'un poisson avec le modèle élastique. Les résultats sont en très bon accord avec ceux obtenus dans le chapitre précédent. Nous avons alors étudié la nage d'un poisson passif (mort) placé dans un sillage généré par la présence d'un demi cylindre. Le poisson se déforme sous l'effet des tourbillons, et, sous certaines conditions favorables, est capable de "nager" pour remonter le courant. Enfin, le serpent de mer a été modélisé et simulé numériquement en deux dimensions. Les résultats sont très satisfaisants. Nous travaillons actuellement sur le modèle en trois dimensions. Ce modèle de poutre élastique devrait également nous permettre de prendre en compte la flexibilité des pales d'éoliennes pour étudier les interactions éventuelles avec le mat.

Finalement, nous nous sommes confrontés aux limites de la méthode de calcul sur grilles cartésienne. En effet, il est impossible de modéliser et calculer les phénomènes de type couches limites en régime turbulent, car ces phénomènes se développent à des échelles très petites devant tout autres quantités d'intérêts comme le sillage. Bien sûr, il est toujours possible de resserrer le maillage proche de ces obstacles où les couches limites apparaissent. On peut imaginer des méthodes de raffinement de type quadtree ou octree. Ce raffinement en "arbre" conserve la topologie des cellules (des carrés ou cubes), et ne permet pas un allongement des cellules comme c'est le cas pour un maillage dit de couche limite. Les pas de temps associés à ces petites mailles seraient réellement trop petits si on utilise une méthode eulérienne. Une alternative est alors les méthodes overset de chimère dans lesquelles nous pouvons modifier la topologie des mailles (sur la grille conforme) et donc approcher un maillage de couche limite. Il s'agit, de ce fait, de la méthode que j'envisage d'étudier pour simuler dans un futur proche les applications en lien avec les éoliennes ou les bateaux.

Chapitre 5

Conclusion générale et perspectives

Mes activités de recherches sont centrées autour d'un point commun : le calcul scientifique. En particulier, je me suis intéressé à des problèmes de modélisation et de simulations numériques en mécanique des fluides. Ces thématiques se sont peu à peu ouvertes à l'étude de problèmes multi physiques. Ces travaux ont pu être réalisés dans d'excellentes conditions dans l'équipe INRIA MC2 et plus largement dans l'équipe calcul scientifique de l'Institut de Mathématiques de Bordeaux, sans oublier les infrastructures informatiques de hautes qualités (clusters de calcul PLAFRIM et AVAKAS). A travers ces travaux j'ai pu développer de nouvelles connaissances en mathématiques appliquées, en calcul scientifique et en physique des écoulements fluides. Ces connaissances ont été traduites numériquement dans le code massivement parallèle NaSCar que j'ai développé depuis 2010. Il est très appréciable notamment de pouvoir partager ces connaissances, et réciproquement, avec le milieu industriel. La collaboration avec la société VALEOL m'a permis de mieux comprendre les besoins de l'entreprise et d'adapter une partie de mes travaux pour mieux répondre à ces besoins. Cette collaboration devrait continuer avec une deuxième thèse cifre.

Lors de tous ces travaux j'ai été confronté aux mêmes problématiques, en particulier les limites de l'utilisation des méthodes développées. On peut aisément conclure que les modèles réduits POD ne sont pas une solution miracle, et que les maillages cartésiens ne sont pas la solution à tous les problèmes. En effet, l'expertise de ces méthodes est nécessaire. Chaque méthode possède ses domaines de validité et d'efficacité. A cet égard, la connaissance des phénomènes physiques étudiés est très précieuse et permet d'apporter une partie des réponses sur l'expertise de ces méthodes. Elle permet également d'apporter des solutions pour élargir les domaines de validité. On peut penser en particulier à la robustesse de la base POD, à la stabilité des modèles réduits, à la validité des calculs tridimensionnels en présence de couches limites, aux modèles de turbulence, *etc.* Même si la POD et les méthodes sur maillages cartésiens présentent les défauts mentionnés ci-dessus, elles restent cependant de très bon choix pour de nombreux problèmes. En particulier les méthodes sur maillage cartésien sont très adaptées pour simuler des écoulements pour des nombres de Reynolds faibles à modérés. Pour des nombres de Reynolds élevés, des précautions doivent être prises. Tout comme pour les modèles réduits POD. On peut modéliser de nombreux écoulements, faire du contrôle, essayer des prédictions, mais en prenant des précautions. Le connaissance physique reste indissociable des problèmes de modélisation et de simulation numérique.

Je vais encadrer une nouvelle thèse (avec Mejdj Azaiez, I2M Bordeaux), en partenariat avec l'entreprise ABENGOA¹ localisée à Séville en Espagne. Dans ce projet nous nous intéresserons à la simulation numérique du processus décrivant le stockage d'énergie par changement de phase. Ceci nécessite la résolution de problèmes tridimensionnels diphasiques avec des interfaces mobiles et à topologie complexes. Les outils que nous avons développés ces dernières années pourraient en partie permettre de d'appréhender numériquement ce type de problème. Par contre, la simulation numérique d'un temps de chargement, c'est à dire le temps de transformer le sel de l'état cristal en liquide, peut prendre quelques jours de calculs. Le but est de travailler sur une alternative de réduction d'ordre avec un double objectifs, la réduction du temps de calculs et la possibilité de varier certains paramètres tels que la conductivité et la résistance de contact voir même la topologie des matrices poreuses contenant le milieu à changement de phase, le sel par exemple. Dans la recherche des modèles réduits, nous envisageons de considérer deux familles réductions de modèles : la PGD (Proper Generalized Decomposition) et la RB (Reduced Basis).

1. <http://www.abengoa.com>

A court terme, nous allons également développer des méthodes pour simuler des phénomènes physiques à différentes échelles. Pour ce faire nous avons besoin d'adapter le maillage à ces types de problèmes. Nous développons actuellement avec OPTIMAD Engineering des méthodes de raffinement de maillage. Ces méthodes sont toujours basées sur l'utilisation de grilles cartésiennes. La première méthode est quadtree/octree. Cette méthode est très facilement parallélisable et permet de simuler finement les sillages par exemple. Par contre, sauf avec une quantité très grande de mailles, elle ne permet pas de simuler des couches limites (turbulentes de surcroît). Nous développons alors des méthodes overset de type chimère dans laquelle se superpose un maillage cartésien classique avec un maillage conforme à l'obstacle (on peut penser à un maillage structuré en "O" pour des géométries pas trop complexes comme des pales d'éoliennes). La génération de ce type de maillage est réalisée grâce aux fonctions level set et au squelette (pour borner la taille que peut avoir ce maillage pour des corps convexes).

A moyen terme, nous allons étudier des écoulements complexes et plus particulièrement des écoulements en micro-canaux chargés en particules. Cette étude se place dans une collaboration naissante avec Annie Colin et Hugues Bodiguel (LOF²-Solvay, Pessac). Plusieurs projets dans cette thématique sont en train d'émerger. Le LOF est particulièrement intéressé la rhéologie d'écoulements avec beaucoup de particules, et également les émulsions. Il peut être possible de modéliser ces fluides de façon mésoscopique. Plusieurs modèles ont été proposés, mais beaucoup de paramètres sont empiriques et peuvent être calibrés. Les méthodes de simulations numériques que nous avons développées se prêtent parfaitement à cette étude. Nos résultats numériques pourront également être validés (et réciproquement) avec les expériences réalisées au LOF. Une application envisagée est le séquençage de l'ADN. Un dispositif novateur est mis en place. Il consiste à placer les brins dans un écoulement en micro canaux pour un écoulement visco-élastique. Le caractère "élastique" est important et permet aux brins de se placer à une certaine distance des bords du canal en fonction de leurs tailles. En fonction de leur position, ils seront entraînés à une vitesse caractéristique qui identifiera leur taille.

Finalement, en conclusion, notre projet de recherche INRIA MC2 a pris fin et nous avons proposé deux nouvelles équipes. Mes thématiques de recherches seront guidées par les thématiques du nouveau projet de notre axe "multi physiques et interactions" (et réciproquement). Ce projet donne entre autre une nouvelle perspective à la réduction de modèles, les origines de mes activités de recherche.

Bibliographie

- R. ABGRALL, H. BEAUGENDRE et C. DOBRZYNSKI : An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257:83–101, 2014.
- D. ADALSTEINSSON et J.A. SETHIAN : The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.
- S. AHMED, S. BAK, J. MCLAUGHLIN et D. RENZI : A Third Order Accurate Fast Marching Method for the Eikonal Equation in Two Dimensions. *SIAM Journal on Scientific Computing*, 33(5):2402–2420, 2011.
- E. AKERVIK, G. HÖPFFNER, U. EHRENSTEIN et D. S. HENNINGSON : Optimal growth, model reduction and control in a separated boundary-layer flow using global eigenmodes. *J. Fluid Mech.*, 579:305–314, 2007.
- S. ALBEN : Simulating the dynamics of flexible bodies and vortex sheets. *J. Comput. Phys.*, 228(7):2587 – 2603, 2009. ISSN 0021-9991.
- P. ANGOT, C.H. BRUNEAU et P. FABRIE : A penalization method to take into account obstacles in a incompressible flow. *Num. Math.*, 81(4):497–520, 1999.
- E. ARQUIS : *Convection mixte dans une couche poreuse verticale non confinée. Application à l'isolation perméodynamique*. Thèse de doctorat, Université Bordeaux I, 1984.
- N. AUBRY, P. HOLMES, J. L. LUMLEY et E. STONE : The dynamics of coherent structures in the wall region of a turbulent boundary layer. *J. Fluid Mech.*, 192:115–173, 1988.
- D.S. BARRETT, M.S. TRIANTAFYLLOU, D.K.P. YUE, M.A. GROSENBAUCH et M.J. WOLFGANG : Drag reduction in fish-like locomotion. *J. Fluid Mech.*, 392:182–212, 1999.
- Y. BAZILEVS, V. M. CALO, J. A. COTTRELL, T. J. R. HUGUES, A. REALI et G. SCOVAZZI : Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 197:173–201, 2007.
- D.N. BEAL, F.S. HOVER, M.S. TRIANTAFYLLOU, J.C. LIAO et Lauder G.V. : Passive propulsion in vortex wakes. *J. Fluid Mech.*, 549:385–402, 2006.
- J-D. BENAMOU et Y. BRENIER : A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik*, 84:375–393, 2000.
- M. BERGMANN : *Optimisation aérodynamique par réduction de modèle POD et contrôle optimal. Application au sillage laminaire d'un cylindre circulaire*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Nancy, France, 2004.
- M. BERGMANN, C.H. BRUNEAU et A. IOLLO : Enablers for robust pod models. *J. Comput. Phys.*, 228(2):516–538, 2009.
- M. BERGMANN, T. COLIN, A. IOLLO, D. LOMBARDI, O. SAUT et H. TELIB : Reduced order models at work in aeronautics and medicine. *Modeling, Simulation and Applications*, 9, 2013.
- M. BERGMANN et L. CORDIER : Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *J. Comp. Phys.*, 227(16):7813–7840, 2008.

- M. BERGMANN, L. CORDIER et J.-P. BRANCHER : Optimal rotary control of the cylinder wake using POD Reduced Order Model. *Phys. Fluids*, 17(9):097101 :1–21, 2005.
- M. BERGMANN, C. GALUSINSKI, A. IOLLO et L. WEYNANS : Drag and drop simulations : from images to full 3d simulations. *The Eighth International Conference on Computational Fluid Dynamics*, July 14-18 2014a. Chengdu, China.
- M. BERGMANN, J. HOVNANIAN et A. IOLLO : An accurate cartesian method for incompressible flows with moving boundaries. *Communications in Computational Physics*, 15(5):1266–1290, 2014b.
- M. BERGMANN et A. IOLLO : Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230(2):329 – 348, 2011.
- M. BERGMANN et A. IOLLO : Numerical simulation of horizontal-axis wind turbine (hawt). *The Seventh International Conference on Computational Fluid Dynamics*, July 9-13 2012. Mauna Lani Bay, Hawaii, USA.
- M. BERGMANN, F. LUDDENS et L. WEYNANS : A sharp cartesian method for the simulation of air water interface. *The Eighth International Conference on Computational Fluid Dynamics*, July 14-18 2014c. Chengdu, China.
- M. BERGMANN, H. TELIB et A. IOLLO : Implementation and validation of hybrid basis function rom method. Future fast aeroelastic simulation technologies, D2.17, 2012a.
- M. BERGMANN, H. TELIB et A. IOLLO : Implementation and validation of hybrid basis function rom method. Future fast aeroelastic simulation technologies, D2.18, 2012b.
- R.W. BLAKE : Fish functional design and swimming performance. *Journal of Fish Biology*, 65:1193–1222, 2004.
- R.W. BLAKE, L.M. CHATTERS et P.D. DOMENICI : Turning radius of yellowfin thuna (*thunnus albacares*) in unsteady swimming manoeuvres. *Journal of Fish Biology*, 46:536–538, 1995.
- I BORAZJANI et F SOTIROPOULOS : Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*, 211(10):1541–1558, 2008.
- I. BORAZJANI et F. SOTIROPOULOS : Why don't mackerels swim like eels? The role of form and kinematics on the hydrodynamics of undulatory swimming. *Physics of Fluids*, 21(9):091109, 2009.
- A. BOUHARGUANE, A. IOLLO et L. WEYNANS : Numerical solution of the monge-kantorovich problem by picard iterations. Inria research report, RR-8477, 2014.
- J.U BRACKBILL, D.B KOTHE et C ZEMACH : A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354, 1992.
- T BRACONNIER, M FERRIER, J-C JOUHAUD, M MONTAGNAC et P SAGAUT : Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers and Fluids*, 40(1):195–209, 2011.
- Y. BRENIER : Polar factorization and monotone rearrangement of vector-valued functions. *Communication in Pure and Applied Mathematics*, 64:375–417, 1991.
- D.L. BROWN, R. CORTEZ et M.L. MINION : Accurate Projection Methods for the Incompressible Navier–Stokes Equations. *Journal of Computational Physics*, 168(2):464–499, 2001.
- M. BUFFONI, S. CAMARRI, A. IOLLO et M.V. SALVETTI : Low-dimensional modelling of a confined three-dimensional wake flow. *J. Fluid Mech.*, 569:141–150, 2006.
- T. BUI-THANH, K. WILLCOX, O. GHATTAS et B. van BLOEMEN WAANDER : Goal-oriented, model-constrained optimization for reduction of large-scale systems. *J. Comp. Phys.*, 224(2):880–896, 2007.
- J. BURKARDT, M. D. GUNZBURGER et H.-C. LEE : Centroidal Voronoi Tessellation-Based Reduced-Order Modeling of Complex Systems. Rapport technique, Florida State University, 2004.

- W. CAZEMIER : *Proper Orthogonal Decomposition and low-dimensionnal models for turbulent flows*. Thèse de doctorat, université de Groningen, 1997.
- W. CAZEMIER, R. W. C. P. VERSTAPPEN et A. E. P. VELDMAN : Proper Orthogonal Decomposition and low-dimensional models for driven cavity flows. *Phys. Fluids*, 10(7):1685–1699, 1998.
- A. CHÉNÉ, C. MIN et F. GIBOU : Second-Order Accurate Computation of Curvatures in a Level Set Framework Using Novel High-Order Reinitialization Schemes. *Journal of Scientific Computing*, 35(2-3):114–131, 2007.
- Y. CHENY et O. BOTELLA : 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(4):1043–1076, 2010.
- A.J. CHORIN : Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.
- P. CINNELLA : High-order residual-based compact schemes on overset grids. *In Workshops schémas aux différences, Méthodes cartésiennes, application en mécanique des fluides*, ENSAM, Paris, 2013.
- D K CLARKE, H A HASSAN et M D SALAS : Euler calculations for multielement airfoils using Cartesian grids. *AIAA Journal*, 24(3):353–358, 1986.
- R. CLIFT, J. R. GRACE et M. E. WEBER : *Bubbles, Drops, and Particles*. Academic Press, New York, 1978.
- C. CONCA, J. SAN MARTIN et M. TUCSNAK : Weak solutions of the equations modelling the motion of a rigid body in a viscous fluid. *Comm. Partial Differential Equations*, 25:1019–1042, 2000.
- M. COQUERELLE et G.H. COTTET : A vortex level set method for the two way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.*, 227(21):9121–9137, 2008.
- L. CORDIER et M. BERGMANN : Proper Orthogonal Decomposition : an overview. *In Lecture series 2002-04 on post-processing of experimental and numerical data*. Von Kármán Institute for Fluid Dynamics, 2002a.
- L. CORDIER et M. BERGMANN : Two typical applications of POD : coherent structures eduction and reduced order modelling. *In Lecture series 2002-04 on post-processing of experimental and numerical data*. Von Kármán Institute for Fluid Dynamics, 2002b.
- F. COUDERC : *Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la désintégration assistée d'un jet liquide par un courant gazeux*. Thèse de doctorat, ENSAE - Toulouse, France, 2007.
- M. COUPLET, C. BASDEVANT et P. SAGAUT : Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *J. Comp. Phys.*, 207:192–220, 2005.
- R.G. COX : The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow. *Journal of Fluid Mechanics*, 168:169–194, 1986.
- J.O. DABIRI, S.P. COLIN, J.H. COSTELLO et M. GHARIB : Flow patterns generated by oblate medusan jellyfish : field measurements and laboratory analyses. *The Journal of Experimental Biology*, 208:1257–1265, 2005.
- H. DAI, H. LUO, J S A Ferreira de SOUSA et J. F DOYLE : Thrust performance of a flexible low-aspect-ratio pitching plate. *Physics of Fluids*, 24(10):101903, 2012.
- B.J. DALY : Numerical Study of Two Fluid Rayleigh-Taylor Instability. *Physics of Fluids*, 10(2):297, 1967.
- B.J. DALY : Numerical Study of Density-Current Surges. *Physics of Fluids*, 11(1):15, 1968.
- A. E. DEANE, I. G. KEVREKIDIS, G. E. KARNIAKAKIS et S. A. ORSZAG : Low-dimensional models for complex geometry flows : Application to grooved channels and circular cylinders. *Phys. Fluids*, 3(10):2337–2354, 1991.
- J. DELVILLE, L. UKEILEY, L. CORDIER, J.-P. BONNET et M. GLAUSER : Examination of large-scale structures in a turbulent mixing layer. Part 1. Proper Orthogonal Decomposition. *J. Fluid Mech.*, 391:91–122, 1999.

- J. DENG, X. M. SHAO et Z. S. YU : Hydrodynamic studies on two traveling wavy foils in tandem arrangement. *Physics of Fluids*, 19:113104, 2007.
- P.D. DOMENICI et R.W. BLAKE : The kinematics and performance of the scape response in the angelfish (*pterophyllum eimekei*). *Canadian Journal of Zoology*, 156:187–205, 1991.
- M. FAHL : *Trust-Region methods for flow control based on Reduced Order Modeling*. Thèse de doctorat, Trier university, 2000.
- R. FEDKIW, T. ASLAM, B. MERRIMAN et S. OSHER : A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999a.
- R. P FEDKIW, T. ASLAM, B. MERRIMAN et S. OSHER : A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999b.
- H. FUJITA et N. SAUER : Construction of weak solutions of the Navier-Stokes equation in a noncylindrical domain. *Bulletin of the American Mathematical Society*, 75(2):465–468, 1969.
- B. GALLETTI, C.-H. BRUNEAU, L. ZANNETTI et A. IOLLO : Low-order modelling of laminar flow regimes past a confined square cylinder. *J. Fluid Mech.*, 503:161–170, 2004.
- M. GAZZOLA, P. CHATELAIN, W.M. van REES et P. KOUMOUTSAKOS : Accepted Manuscript. *Journal of Computational Physics*, pages 1–48, mai 2011.
- R GHAS, R MITTAL et H DONG : A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics*, 225(1):528–553, 2007.
- F. GIBOU, R. FEDKIW, L.-T. CHENG et M. KANG : A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains. *Journal of Computational Physics*, 176(1):205–227, 2002.
- R. GLOWINSKI, T.-W. PAN, T.I. HESLA, D.D. JOSEPH et J. PÉRIAUX : A fictitious domain approach for the direct simulation of incompressible fluid flow past moving rigid bodies : Application to particulate flow. *J. Comp. Phys.*, 169:363–426, 2001.
- K. GODA : A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of Computational Physics*, 30(1):76 – 95, 1979.
- S. GOTTLIEB et C.-W. SHU : Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society*, 67(221):73–85, 1998.
- W. R. GRAHAM, J. PERAIRE et K. T. TANG : Optimal Control of Vortex Shedding Using Low Order Models. Part 2 : Model-based control. *Int. J. for Numer. Meth. in Engrg.*, 44(7):973–990, 1999b.
- F. H HARLOW et J.E. WELCH : Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8(12):2182, 1965.
- A. HARTEN, B. ENGQUIST, S. OSHER et S.R. CHAKRAVARTHY : Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71(2):231–303, 1987.
- P. HOLMES, J. L. LUMLEY et G. BERKOOZ : *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics, 1996.
- J. HOVNANIAN : *Méthode de Frontières Immergées pour la Mécanique des Fluides. Application à la Simulation de la Nage*. Thèse de doctorat, Université de Bordeaux 1, Bordeaux, France, 2012.
- T. J. R. HUGUES, G. FEIJÓO, L. MAZZEI et J. B. QUINCY : The variational multiscale method-a paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166:3–24, 1998.
- A. IOLLO, A. DERVIEUX, J. A. DÉSIÉRI et S. LANTERI : Two stable pod-based approximations to the navier-stokes equations. *Comput. Visual. Sci.*, 3:61–66, 2000a.
- A. IOLLO, S. LANTERI et J. A. DÉSIÉRI : Stability properties of pod-galerkin approximations for the compressible navier-stokes equations. *Theoret. Comput. Fluid Dynamics*, 13:377–393, 2000b.

- K. ITO et S. S. RAVINDRAN : A reduced-order method for simulation and control of fluid flows. *J. Comp. Phys.*, 143:403–425, 1998.
- M. JEMISON, E. LOCH, M. SUSSMAN, M. SHASHKOV, M. ARIENTI, M. OHTA et Y. WANG : A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54 (2-3):454–491, 2013.
- G.-S. JIANG et D. PENG : Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM Journal on Scientific Computing*, 21(6):2126–2143, 2000.
- G.-S. JIANG et C.-W. SHU : Efficient implementation of weighted {ENO} schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.
- G. JIN et M. BRAZA : A Nonreflecting Outlet Boundary Condition for Incompressible Unsteady Navier-Stokes Calculations. *J. Comp. Phys.*, 107(2):239–253, 1993.
- X. JIN, B. GAILLARDON, M. BERGMANN et A. IOLLO : A simplified design tool for large wind turbine blades. *The European Wind Energy Association conference*, April 16-19 2012. Copenhagen, Danemark.
- V. L. KALB et A. E. DEANE : An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models. *Phys. Fluids*, 19:054106, 2007.
- M. KANG, R. P FEDKIW et X.-D. LIU : A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.
- G. S. KARAMANOS et G. E. KARNIADAKIS : A spectral vanishing viscosity method for Large Eddy Simulations. *J. Comp. Phys.*, 162:22–50, 2000.
- H.-J. KULL : Theory of the Rayleigh-Taylor instability. *Physics Reports*, 206(5):197–325, 1991.
- J. LEE, J. KIM, H. CHOI et K.-S. YANG : Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *Journal of Computational Physics*, 230(7):2677–2695, 2011.
- G.-J. LI et X.-Y. LU : Force and power of flapping plates in a fluid. *Journal of Fluid Mechanics*, 712:598–613, octobre 2012.
- J.C.. LIAO, D.N. BEAL, G.V. LAUDER et M.S. TRIANTAFYLLOU : Fish exploiting vortices decrease muscle activity. *Science*, 302:1566–1569, 2003.
- M.J. LIGHTHILL : Note on the swimming of slender fish. *J. Fluid Mech.*, 9:305–317, 1960.
- M.J. LIGHTHILL : Hydromechanics of aquatic propulsion : a survey. *Annual Review of Fluid Mechanics.*, 1:413–446, 1969.
- M.J. LIGHTHILL : Aquatic animal propulsion of high hydrodynamical efficiency. *J. Fluid Mech.*, 44:265–301, 1970.
- M.J. LIGHTHILL : Large amplitude elongated-body theory of fish locomotion. *Proc. R. Soc. Mech. B*, 179:125–138, 1971.
- C.C. LINDSEY : *Form, function and locomotory habits in fish*. W.S. Hoar and D.J. Randall, Eds. New-York : Academic, 1978.
- X.-D. LIU, R. P FEDKIW et M. KANG : A boundary condition capturing method for Poisson’s equation on irregular domains. *Journal of Computational Physics*, 160(1):151–178, 2000.
- X-D. LIU, S. OSHER et T. CHAN : Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.
- E. LOMBARDI, M. BERGMANN, S. CAMARRI et A. IOLLO : Low-order models : Optimal sampling and linearized control strategies. *Journal Europeen des Systemes Automatises*, 45(7-10):575–593, 2011.

- J. L. LUMLEY : *Atmospheric Turbulence and Wave Propagation. The structure of inhomogeneous turbulence*, pages 166–178. A.M. Yaglom & V.I. Tatarski, 1967.
- X. MA et G. E. KARNIADAKIS : A low-dimensional model for simulating three-dimensional cylinder flow. *J. Fluid Mech.*, 458:181–190, 2002.
- O. MARSDEN : On the use of high-order explicit finite differences in bounded aeroacoustic computations. *In Workshops schémas aux différences, Méthodes cartésiennes, application en mécanique des fluides*, ENSAM, Paris, 2013.
- T MCMILLEN et P HOLMES : An elastic rod model for anguilliform swimming. *Journal of Mathematical Biology*, 53(5):843–886, 2006.
- R MITTAL, H DONG, M BOZKURTAS, F M NAJJAR, A VARGAS et A von LOEBBECKE : A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825–4852, 2008.
- R. MITTAL et G. IACCARINO : Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- G. MONGE : Memoire sur la theorie des des déblais et des remblais. *Histoire de l'Academie des Sciences de Paris*, 1781.
- Š. NEČASOVÁ, T. TAKAHASHI et M. TUCSNAK : Weak solutions for the motion of a self-propelled deformable structure in a viscous incompressible fluid. *Acta applicandae mathematicae*, 116(3):329–352, 2011.
- B. R. NOACK, K. AFANASIEV, M. MORZYŃSKI, G. TADMOR et F. THIELE : A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.*, 497:335–363, 2003.
- B. R. NOACK et H. ECKELMAN : A low dimensional galerkin method for the three-dimensional flow around a circular cylinder. *Phys. Fluids*, 6(1):124–143, 1994.
- B. R. NOACK, P. PAPAS et P. A. MONKEWITZ : The need for a pressure-term representation in empirical galerkin models of incompressible shear-flows. *J. Fluid Mech.*, 523:339–365, 2005.
- B. R. NOACK, M. SCHLEGEL, B. AHLBORN, G. MUTSCHKE, M. MORZYNSKI, P. COMTE et G. TADMOR : A finite-time thermodynamics formalism for unsteady flows. *J. Non-Equilib. Thermodyn.*, 3(2):103–148, 2008.
- J. NOCEDAL et S. J. WRIGHT : *Numerical Optimization*. Springer series in operations research, 1999.
- S. OSHER et J. A. SETHIAN : Fronts propagating with curvature-dependent speed : Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(12), 1988.
- C.S. PESKIN : Flow patterns around heart valves : A numerical method. *J. Comp. Phys.*, 10:252–275, 1972.
- J. PINILLA VELANDIA : *Modélisation et simulation à l'échelle du pore de la récupération assistée des hydrocarbures par injection de polymers*. Thèse de doctorat, Université de Bordeaux 1, 2012.
- B. PODVIN et J. LUMLEY : A low-dimensional approach for the minimal flow unit. *J. Fluid Mech.*, 362:121–151, 1998.
- R. D. PRABHU, S. S. COLLIS et Y. CHANG : The influence of control on Proper Orthogonal Decomposition of wall-bounded turbulent flows. *Phys. Fluids*, 13(2):520–537, 2001.
- S. S. RAVINDRAN : A reduced-order approach for optimal control of fluids using Proper Orthogonal Decomposition. *Int. J. Numer. Meth. Fluids*, 34:425–448, 2000.
- D. REMPFER : Investigations of boundary layer transition via Galerkin Projections on Empirical Eigenfunctions. *Phys. Fluids*, 8(1):175–188, 1996.
- D. REMPFER : On low-dimensional Galerkin models for fluid flow. *Theor. Comput. Fluid Dyn.*, 14:75–88, 2000.

- D. REMPFER et H. F. FASEL : Evolution of three-dimensional coherent structures in a flat-plate boundary layer. *J. Fluid Mech.*, 260:351–375, 1994.
- C.M. RHIE et W.L. CHOW : Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, novembre 1983.
- C. W. ROWLEY : Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. on Bifurcation and Chaos*, 15(3):997–1013, 2005.
- G. RUSSO et P. SMEREKA : A Remark on Computing Distance Functions. *Journal of Computational Physics*, 163(1):51–67, 2000.
- Y. SAAD et M. H. SCHULTZ : Gmres : A generalized residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.
- P. SAGAUT : *Large-eddy simulation for incompressible flows - An introduction*. Springer-Verlag, 2005.
- J. SAN MARTIN, J.-F. SCHEID, T. TAKAHASHI et M. TUCSNAK : An initial and boundary value problem modeling of fish-like swimming. *Arch. Rational Mech. Anal.*, 188:429–455, 2008.
- J. SAN MARTIN et V. STAROVOITOV : Global weak solutions for the two dimensional motion of several rigid bodies in an incompressible viscous fluid. *Archive for Rational Mechanics and Analysis*, 161:113–147, 2002.
- V. K. SAUL'EV : On solving boundary value problems with high performance computers by a fictitious domain method. *Siberian Mathematical journal*, page 912, 1963.
- J. A. SETHIAN : *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, 1999.
- J.A. SETHIAN : A fast marching level set method for monotonically advancing fronts. *Applied Mathematics*, 93:1591–1595, 1996.
- M. SFATIOTAKIS, D.M. LANE et Bruce D.J. : Review of fish swimming modes for aquatic locomotion. *IEEE J. Oceanic Eng.*, 24(2):237–252, 1999.
- C.-W. SHU et S. OSHER : Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471, 1988.
- S. SIRISUP et G. E. KARNIADAKIS : A spectral viscosity method for correcting the long-term behavior of POD model. *J. Comp. Phys.*, 194:92–116, 2004.
- S. SIRISUP, G. E. KARNIADAKIS, D. XIU et I. G. KEVREKIDIS : Equation-free/galerkin-free pod-assisted computation of incompressible flows. *J. Comp. Phys.*, 207:568–587, 2005.
- L. SIROVICH : Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics*, XLV (3):561–590, 1987.
- J.N. SØRENSEN : Aerodynamic Aspects of Wind Energy Conversion. *Annual Review of Fluid Mechanics*, 43 (1):427–448, janvier 2011.
- J.A. SPARENBERG : Survey of the mathematical theory of fish locomotion. *Journal of Engineering Mathematics*, 44:395–448, 2002.
- S. STÖCKER et D. WEIHS : Optimization of energetic advantages of burst swimming of fish. *Mathematical Methods in the Applied Sciences*, 24:1387–1400, 2001.
- M. SUSSMAN : A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110–136, 2003.
- M. SUSSMAN et E. FATEMI : An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

- M. SUSSMAN et E.G. PUCKETT : A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, 2000.
- M. SUSSMAN, P. SMEREKA et S. OSHER : A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146 – 159, 1994.
- S. TANGOONE : *Calcul numérique et stabilité d'écoulements diphasiques tridimensionnels en microfluidique*. Thèse de doctorat, Université de Bordeaux 1, Bordeaux, France, 2007.
- R. TEMAM : Sur l'approximation de la solution des equations de navier-stokes par la méthode des pas fractionnaires ii. *Archiv. Rat. Mech. Anal.*, 32:377–385, 1969.
- L.J.P. TIMMERMANS, P.D. MINEV et F.N. VAN DE VOSSE : An approximate projection scheme for incompressible flow using spectral elements. *International journal for numerical methods in fluids*, 22(7):673–688, 1996.
- J.J. VIDELER et D. WEIHS : Energetic advantages of burst-and-coast swimming of fish at high speeds. *J. exp. Biol.*, 97:169–178, 1982.
- G. VIGO : The Proper Orthogonal Decomposition applied to unsteady compressible Navier-Stokes equation. Rapport technique 3945, INRIA, 1998.
- C. VILLANI : *Topics in optimal transportation*. American Mathematical Society, 1st édition, 2003.
- C. VILLANI : *Optimal Transport, old and new*. Springer-Verlag, 1st édition, 2009.
- A. von LOEBBECKE, R. MITTAL, F. FISH et R. MARK : Propulsive efficiency of the underwater dolphin kick in humans. *J Biomech Eng.*, 131(5):054504, 2009.
- A.W. VREMAN : An eddy-viscosity subgrid-scale model for turbulent shear flow : Algebraic theory and applications. *Physics of Fluids*, 16(10):3670–3681, 2004.
- P.W. WEBB : Body form, locomotion and foraging in aquatic vertebrates. *American zoologist*, 24:107–120, 1984.
- P.W. WEBB et R.S. KEYES : Division of labour between median fins in swimming dolphins (pisces : Coryphaeidae). *Copeia*, pages 901–904, 1981.
- D. WEIHS : Energetic advantages of burst swimming of fish. *J. theor. Biol.*, 48:215–229, 1974.
- D. WEIHS et P.W. WEBB : *Optimization of locomotion*. P.W. Webb and D.Weihs, Eds. New-York : Praeger, 1983.
- K. WILLCOX et J. PERAIRE : Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.
- T. YE, R. MITTAL, H.S. UDAYKUMAR et W. SHYY : An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209–240, 1999.
- C.-L. YU, S.-C. TING, M.-K. YEH et J.-T. YANG : Three-dimensional numerical simulation of hydrodynamic interactions between pectoral-fin vortices and body undulation in a swimming fish. *Physics of Fluids*, 23(9):091901, 2011.
- Q. ZHU, M.J. WOLFGANG, D.K.P. YUE et M.S. TRIANTAFYLLOU : Three-dimensional flow structures and vorticity control in fish-like swimming. *J. Fluid Mech.*, 468:1–28, 2002.